GE Fanuc Automation

Computer Numerical Control Products

# Series 16 / 18 / 20 / 21
# Macro Compiler / Macro Executor

## Programming Manual

GFZ-61803E-1/07 August 1997

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.

This document is based on information available at the time of its publication. While efforts have been made to be accurate, the information contained herein does not purport to cover all details or variations in hardware or software, nor to provide for every possible contingency in connection with installation, operation, or maintenance. Features may be described herein which are not present in all hardware and software systems. GE Fanuc Automation assumes no obligation of notice to holders of this document with respect to changes subsequently made.

GE Fanuc Automation makes no representation or warranty, expressed, implied, or statutory with respect to, and assumes no responsibility for the accuracy, completeness, sufficiency, or usefulness of the information contained herein. No warranties of merchantability or fitness for purpose shall apply.

**©Copyright 1997 GE Fanuc Automation North America, Inc.**
**All Rights Reserved.**

# Table of Contents

## PROGRAMMING

# APPENDIX

# PROGRAMMING

# 1 OUTLINE

NC programs include those which are prepared by custom macro and very seldom altered and those which may differ from one another according to relevant machining such as part programs. A batch handling of these programs of different characters may cause 1 quicker battery consumption, or spoil the custom macro in case of misoperation.

To solve a problem, this function will convert the custom macro prepared by a machine tool builders into an execution format, register it to the ROM module, and enables it to be executed.

(1) Since the custom macro is converted into an execution format and registered, the execution speed is high. This will shorten the machining time and improve the machining accuracy.

(2) The registration to the ROM eliminates battery consumption and prevents custom macro damage through misoperation. This will improve the reliability.

(3) Since the registered program is not indicated on the program display, the machine tool builder's knowhow can be protected.

(4) Since the custom macro is registered in the ROM, the program edit memory can effectively be used.

(5) The user can call the macro with an easy call procedure without being conscious of the registered program. On the program edit memory, custom macros can be prepared and executed in the standard manner.

(6) The user can confirm the operation using RAM module before making a ROM. (excluding Series 21–B)

(7) Conversational macro function can compile machine tool builders original screen.

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

| Model name | Abbreviation | | |
|---|---|---|---|
| FANUC Series 16–TA | 16–TA | Series 16–A | Series 16 |
| FANUC Series 16–MA | 16–MA | | |
| FANUC Series 16–TTA | 16–TTA | | |
| FANUC Series 16–GCA | 16–GCA | | |
| FANUC Series 16–GSA | 16–GSA | | |
| FANUC Series 16–PA | 16–PA | | |
| FANUC Series 16–LA | 16–LA | | |
| FANUC Series 16–TB | 16–TB | Series 16–B | |
| FANUC Series 16–MB | 16–MB | | |
| FANUC Series 16–PB | 16–PB | | |
| FANUC Series 16–LB | 16–LB | | |
| FANUC Series 160–TB | 160–TB | | |
| FANUC Series 160–MB | 160–MB | | |
| FANUC Series 160–PB | 160–PB | | |
| FANUC Series 160–LB | 160–LB | | |
| FANUC Series 16–TC | 16–TC | Series 16–C | |
| FANUC Series 16–MC | 16–MC | | |
| FANUC Series 16–PC | 16–PC | | |
| FANUC Series 160–TC | 160–TC | | |
| FANUC Series 160–MC | 160–MC | | |
| FANUC Series 160–PC | 160–PC | | |
| FANUC Series 16*i*–TA | 16*i*–TA | Series 16*i*–A | |
| FANUC Series 16*i*–MA | 16*i*–MA | | |
| FANUC Series 16*i*–PA | 16*i*–PA | | |
| FANUC Series 16*i*–LA | 16*i*–LA | | |
| FANUC Series 160*i*–TA | 160*i*–TA | | |
| FANUC Series 160*i*–MA | 160*i*–MA | | |
| FANUC Series 160*i*–PA | 160*i*–PA | | |
| FANUC Series 160*i*–LA | 160*i*–LA | | |
| FANUC Series 18–TA | 18–TA | Series 18–A | Series 18 |
| FANUC Series 18–MA | 18–MA | | |
| FANUC Series 18–TTA | 18–TTA | | |
| FANUC Series 18–GCA | 18–GCA | | |
| FANUC Series 18–GSA | 18–GSA | | |
| FANUC Series 18–PA | 18–PA | | |
| FANUC Series 18–TB | 18–TB | Series 18–B | |
| FANUC Series 18–MB | 18–MB | | |
| FANUC Series 18–PB | 18–PB | | |
| FANUC Series 180–TB | 180–TB | | |
| FANUC Series 180–MB | 180–MB | | |
| FANUC Series 180–PB | 180–PB | | |

| Model name | Abbreviation | | |
|---|---|---|---|
| FANUC Series 18–TC | 18–TC | Series 18–C | Series 18 |
| FANUC Series 18–MC | 18–MC | | |
| FANUC Series 18–PC | 18–PC | | |
| FANUC Series 180–TC | 180–TC | | |
| FANUC Series 180–MC | 180–MC | | |
| FANUC Series 180–PC | 180–PC | | |
| FANUC Series 18*i*–TA | 18*i*–TA | Series 18*i*–A | |
| FANUC Series 18*i*–MA | 18*i*–MA | | |
| FANUC Series 18*i*–PA | 18*i*–PA | | |
| FANUC Series 180*i*–TA | 180*i*–TA | | |
| FANUC Series 180*i*–MA | 180*i*–MA | | |
| FANUC Series 180*i*–PA | 180*i*–PA | | |
| FANUC Series 20–FA | 20–FA | Series 20–A | Series 20 |
| FANUC Series 20–TA | 20–TA | | |
| FANUC Series 21–TB | 21–TB | Series 21–B | Series 21 |
| FANUC Series 21–MB | 21–MB | | |
| FANUC Series 210–TB | 210–TB | | |
| FANUC Series 210–MB | 210–MB | | |
| FANUC Series 21*i*–TA | 21*i*–TA | Series 21*i*–A | |
| FANUC Series 21*i*–MA | 21*i*–MA | | |
| FANUC Series 210*i*–TA | 210*i*–TA | | |
| FANUC Series 210*i*–MA | 210*i*–MA | | |

The 21–TB model is available in two types: control unit type A and control unit type B. The two types may also be referred to as the 21–TB (control unit A) and the 21–TB (control unit B) when the descriptions of the types differ.

In this manual, the models may be classified as indicated below.

| System | | Model name |
|---|---|---|
| Lathe system | T series | 16–TA, 16–TTA, 16–GCA, 16–TB, 160–TB, 16–TC, 160–TC, 16*i*–TA, 160*i*–TA, 18–TA, 18–TTA, 18–GCA, 18–TB, 180–TB, 18–TC, 180–TC, 18*i*–TA,180*i*–TA, 20–TA, 21–TB, 210–TB, 21*i*–TA, 210*i*–TA |
| Machining center system | M series | 16–MA, 16–GSA, 16–PA, 16–LA, 16–MB, 16–PB, 16–LB, 160–MB, 160–PB, 160–LB, 16–MC, 16–PC, 160–MC, 160–PC, 16*i*–MA, 16*i*–PA, 16*i*–LA, 160*i*–MA, 160*i*–PA, 160*i*–LA, 18–MA, 18–GSA, 18–PA, 18–MB, 18–PB, 180–TB, 180–MB, 180–PB, 18–MC, 18–PC, 180–MC, 180–PC, 18*i*–MA, 18*i*–PA, 180*i*–MA, 180*i*–PA, 20–FA, 21–MB, 210–MB, 21*i*–TA, 210*i*–TA |

The words used in the explanation are defined as follows.

”P-CODE program” :

Execution type macro program prepared by a machine tool builder, being compiled and registered to ROM.

”Execution macro” :

Program to operate machine in P-CODE program.

”Auxiliary macro” :

Program to make an auxiliary operation for the execution macro and the conversational macro in P-CODE program.

”Conversational macro” :

Program to operate screen in P-CODE program.

”User program” :

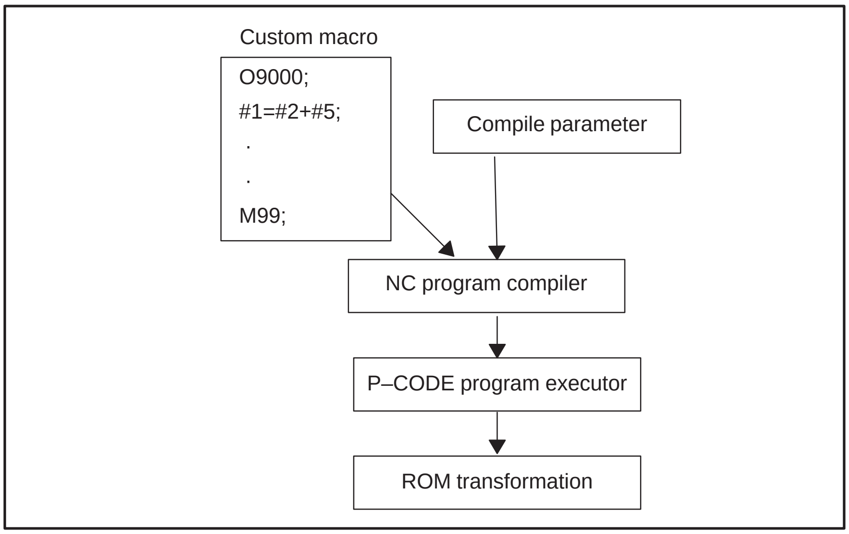
Program prepared by end-user for program edit memory.

# 2 MACRO COMPILER AND MACRO EXECUTOR

## 2.1 MACRO COMPILER

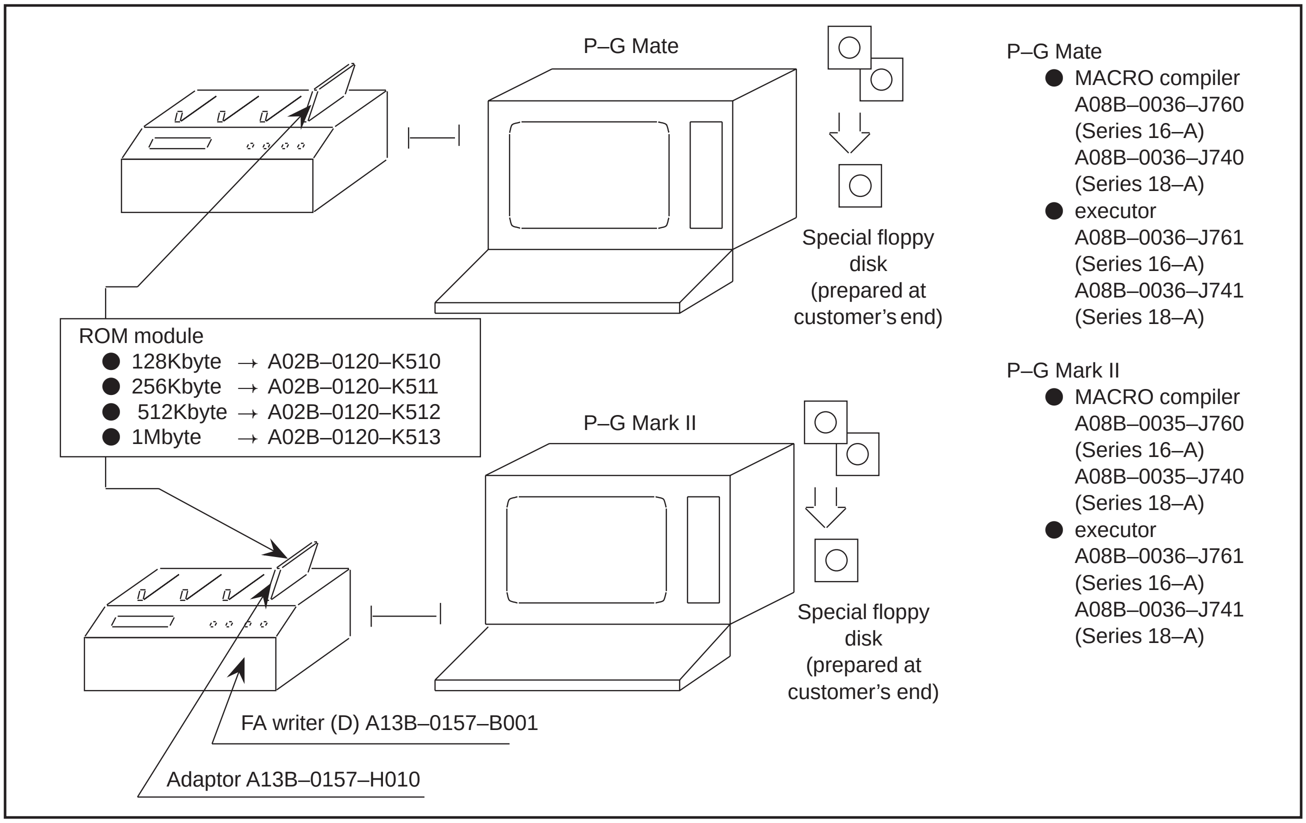
**NOTE**
For the macro compiler for other than Series 16/18–A, refer to the ”FAPT MACRO COMPILER (For Personal Computer) PROGRAMMING MANUAL (B–66102E).” (When reading the manual, skip Section 2.1.)

The NC program is converted into an execution form (P-CODE program), output to the ROM module, and registered into a ROM. Mount the prepared ROM module on the Series 16-A or Series 18 to execute the registered P-CODE program to be called from the user program by means of G, M and T codes or specified code set by parameter.




### 2.1.1 Equipment Needed for Compile

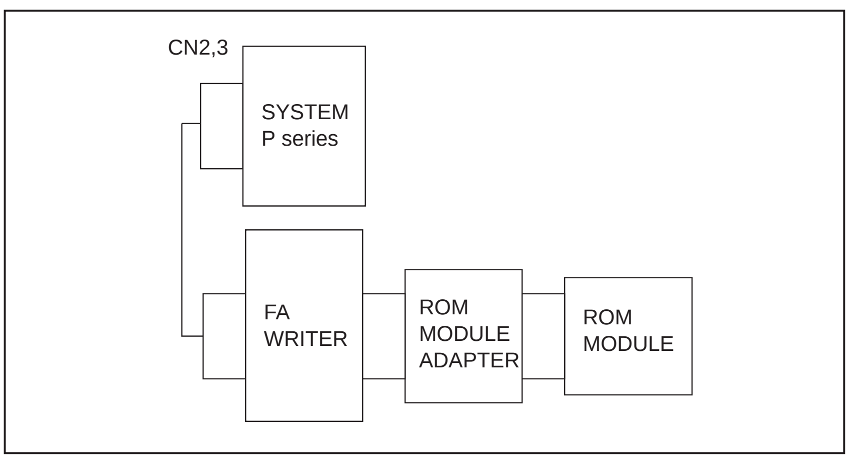
MACRO Compiler Developing Equipment (When SYSTEM P series is used)




**NOTE**
Refer to Appendix D for macro executor with graphic conversation.

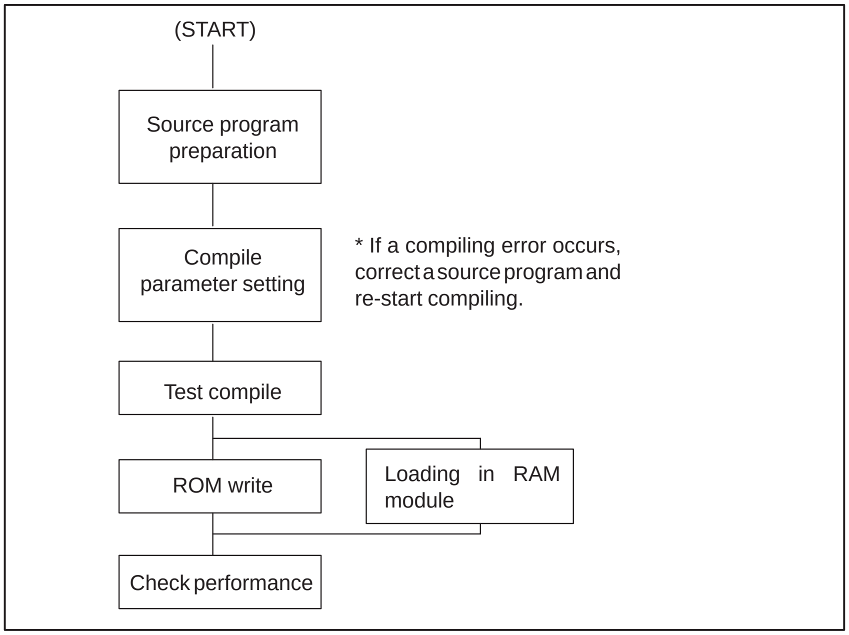
## 2.1.2 Equipment Connection

Connect FA writer to CN2 or CN3 for SYSTEM P series.




**NOTE**
Use CN1 normally for PPR.

## 2.1.3 Compiling Procedure (Main Flow)




## 2.1.4 Compiling Procedure Using System P Series (Details)

(1) Equipment connection

Connect FA writer to CN2 or CN3 of SYSTEM P series. Generally, connect FANUC PPR to CN1.

(2) Turning on power of SYSTEM P series

Turn SYSTEM P series power ON.

(3) Loading system

1. Load the FAPT MACRO compiler system disk to either drive unit.
2. Keep pressing the ”LOAD” key on the left upper side of the keyboard for a few seconds.
3. When the menu is displayed, loading operation is completed.

(4) Source program input and correction

For detailed operation method, refer to APPENDIX B.

1. ”R2” (Display and edit)
2. No.= ”1” (Custom MACRO program display and edit) <**NL**>

(a) Input from keyboard

- ” <**FO**> = OFF, <**F1**> = OFF, <**F2**> = OFF”
- PROGRAM = ”IN” <**NL**>
- INPUT = ”Oxxxx” <**NL**>,”
  Date <**NL**> , , , ,
  only ”<**NL**>” at the last

(b) Reading from floppy disk

- ”<**F0**> = OFF, <**F1**> = OFF, <**F2**> = OFF”

- PROGRAM = ”IN” **<NL>**
- FD = ”OK_ @File Name **<NL>**
  or ”OK _: File No. **<NL>**
  ”_” : Space

(c) Progran correction within memory

- ”**<F0>** = ON, **<F1>** = OFF, **<F2>** = OFF”
- PROGRAM = ”Oxxxx” **<NL>**
- Correct a program, using a screen editor.

(5) Setting Compile parameter

Not required when the compile parameter is already set.

1. For no initial screen, press **<NL>** a few times.
2. ”R1” (setting)
   NO.= ”1” (Parameter) **<NL>**
3. No.= ”1” (parameter setting) **<NL>**
4. Set the parameter, using the screen editor with ”CHG”.
5. ”R0” (End) when the setting of all parameters is completed
6. Only **<NL>** (End of program)

(6) Test compile

1. For no initial screen, press **<NL>** a few times.
2. ”R0” (Start)
3. ”**<F3>** = ON ” : Displays a source program during compile.
   ”**<F3>** = OFF” : Displays o[NL]y program No. during compile.
4. No. = ”1” (Test compile) **<NL>**
5. If an error occurs during compiling, correct the error and compile a program.

(7) Setting FA writer Channel

1. ”R1” (setting)
2. No.= ”2 ” (ROM writer) **<NL>**
3. The current ROM writer channel setting conditions are displayed on the CRT screen.
4. CN1 = : ”ON” **<NL>** : The channel is used.
   : ”OFF” **<NL>** : The channel is not used.
   : only **<NL>** : No setting is changed.
5. CN2 = : Same as above.
6. CN3 = : Same as above.
7. BUILT-IN ROM WRITER =
   : Same as above (only P-G mate)
8. No. = **<NL>** (End)

(8) Preparations for FA writer

1. Install the erased ROM module to the FA writer.
2. Turn the FA writer power ON.
3. Set the FA writer to the remote mode.

(9) ROM writing

1. For no initial screen, press **<NL>** a few times.
2. ”R0” (start)
3. ”**<F7>** = OFF, F9 = ON” : (FA writer selection)
4. ”**<F3>** = ON ” : Displays a source program during compiling.
   ”**<F3>** = OFF” : Displays only program No. during compiling.
5. No. = : ”2” : (ROM writing) **<NL>**
6. Data writing to ROM is started.
7. Writing is completed within a few minutes to 10 minutes or so.
   - If compile parameter ”9000.0 = 0” is selected, no comparison after ROM writing is made, thus making it possible to reduce the writing time.

(10) Performance Check
Install ROM module after writing to ”MACRO” socket on the main CPU PCB of Series 16/ Series 18 to check the operation.
PWE = 1 of setting No.0 :
When power is ON while pressing ”DELETE” key, the 10, 000 level P-CODE variables and program within RAM are cleared.

## 2.1.5 Table for Editing by P–G Mate

| Job | | Command | Keyboard input (enter NL at end of command) | Notes |
|---|---|---|---|---|
| Major classification | Minor classification | | | |
| Data input | Paper tape, keyboard, floppy disk | INPUT | IN [ ␣ a string] | Input terminated by the string specified |
| | Addition from keyboard | KEYIN | K␣ OLD, line number, increment | Data added several line at a time |
| Data display | Display filenames for files on floppy disk | FDLIST | FDL | |
| | Any line | LIST | L[ ␣ line number] | Display from the specified line number |
| | Lines containing the specified string | LIST | L ␣ d string d | |
| Data output | Punch spece | SPACE | SP ␣ n | Punch n speces |
| | Punch feed | FEED | FE ␣ n | Punch n feeds |
| | Output to paper tape, floppy disk | OUTPUT | OUT[ ␣ a string] | |
| Replacement | Alter a whole line | ALTER | Line number ␣1 line of data | |
| | Alter part of a line | ALTER | A ␣ d string1 d [ , d string2 d ], { n ALL } | Replace ALL or n strings with string1 |
| | Replace address character | REPLACE | | Replace character1 with character2 |
| | Reverse order of 2 address data characters | CHANGE | R␣ character1/character2<br>C␣ character1/character2 | Change the order in one block |
| Insertion | Insert1 line of data | INSERT | Line number ␣1 line of data | |
| | Insert string | INSERT | ␣ d string1 d ,<br>[, d string2 d ] { n ALL } | Insert string2 after string1 |
| | Copy and add a specified block | MOVE | MOV[line number1, line number2] | Copy and add the data from line number1 to line number 2 |

| Job: Major classification | Job: Minor classification | Command | Keyboard input (enter NL at end of command) | Notes |
|---|---|---|---|---|
| Deletion | Deletion n whole lines | DELETE | Line number1 [,line number2] | Delet the data from line number1 to line number2 |
| | Delete string | DELETE | D␣ d character d, { n / ALL } | |
| | Delete address data | ADELETE | AD␣ address character | |
| | Delete lines containing a particular string | BDELETE | BD␣ atring | The string can be an address character |
| Scaling to require values | Multiple of a pecific address | SCALE | S ␣ address character / n | n is the scaling factor |
| | Multiple of incremental NC data | ASCALE | AS␣ address character / n | n is the scalling factor |
| Adding and sorting sequence numbers | | SEQNO | SEQ ␣ intial value [ , increment [ , n ]] | If n is given, add a sequence number every n blocks |
| Copy paper tape | | COPY | COPY | Copies any sort of data on paper tape. |
| Modification support | Renumber lines | RENUMBER | REN[ ␣ intial value [ , increment]] | |
| | Change character used to display EOB | EOB | EOB␣ Character | |
| | Display list of commands | HELP | H | |
| | NC data TH, TV check | THTV | THTV | Read in NC data from tape reader, and check TH, TV' |
| | Advance pointer | FIND | F␣ { n / d string d } [,n] | Advance pointer by n lines Advance to a line containing a particular string |
| | Move pointer back | RACK | Bn | Move pointer back n lines. If n is omitted, move pointer back to preceding line |
| | Comment | * | * comment string | Insert any commnet after * |
| Process control | Start editing | EDIT | ED␣ { FAPT / NC } [, { EIA / ISO } ] | |
| | Change data type and code system | MODE | M ␣ { FAPT / NC } [, { EIA / ISO } ] | |
| | End editing | END | E | |
| Special conversion | Change from integer NVC data to floating point NC data | POINT | PO␣ $X_1$ / $n_1$ [ , $X_i$ / $n_i$ ] $_{i=2}^{5}$ | |
| | Change from floating point NC data to integer NC data | INTEGER | INT␣ $X_1$ / $n_1$ [ , $X_i$ / $n_i$ ] $_{i=2}^{5}$ | |
| | ADD a specified amount | ADD | ADD␣ $X_1$ / $n_1$ [ , $X_i$ / $n_i$ ] $_{i=2}^{5}$ | |

## 2.1.6 P–CODE Loader Function

The P-CODE loader function transfers files in Motorola format S, which is a ROM format, to the RAM module installed in the Series 16/18-A or the Series 16-TTA (tool posts 1 and 2). The files are created by the macro compiler for the FANUC SYSTEM P-MODEL G (referred to hereafter as the P-G) or a personal computer (referred to hereafter as the PC). The P-CODE loader function also operates the macro executor in the RAM.

The RS-232C interface is used to connect the Series 16/18–A to the P-G or PC.

The communication parameters for transferring the file in the ROM format (referred to below as loading) depend on the parameters of the Series 16/18–A. The parameters must be set before loading.

If the CAN and PROG keys are pressed simultaneously on the MDI, and the power to the Series 16/18–A is turned off, this function displays the data transfer screen (loading screen). For the Series 16/18-TTA, the function displays the data transfer selection screen. When tool post 1 or 2 is selected on the selection screen, the loading screen is displayed. Data can only be transferred while the loading screen is displayed.

Issue a data transfer command in the P-G or PC. For a description of the transfer commands and other details, refer to the specifications of the FAPT Macro Compiler.

The function eliminates the need to store data on a ROM chip when developing a macro program.

The P-CODE loader function is executed while a power-on sequence is suspended. The functions of the CNC are disabled while the data transfer selection screen or loading screen is displayed. The P-CODE loader function requires an executor option and reader/punch control option.

### 2.1.6.1 Operation

(1) The communication parameters for loading depend on the parameters of the Series 16/18–A. Specify parameters such as the channels to be used and the baud rate before loading.

(2) Turn off the power to the Series 16 and install a RAM module instead of the ROM module for custom macros. In Series 16/18-TTA, install the RAM module(s) in tool post 1 and/or tool post 2. Connect the Series 16/18 to the P-G or PC via the RS-232C interface.

(3) Turn on the power to the Series 16/18 while simultaneously pressing the CAN and PROG keys on the MDI. Hold down the CAN and PROG keys until the title, MACRO COMPILER/EXECUTOR P-CODE LOADER is displayed.

(4) On the Series 16/18-TTA, however, the following data transfer selection screen is displayed instead of the above title.
Pressing address key M on the screen selects tool post 1 and pressing address key S selects tool post 2.
Pressing numeric key 0 starts the Series 16/18-TTA and enables the executor to execute the macros loaded in the RAM.

```
MACRO COMPILER/EXECUTOR P–CODE LOADER

(I/O PARAMETERS)           (DATA TRANSFER)

I/O CHANNEL   = 0          (MAPPING – 1MB)
I/O UNIT      = 0
BAUDRATE      =11
STOP BIT      = 1
INPUT CODE    = 1

LOADING PATH 1  : YES (PUSH M KEY)
LOADING PATH 2  : YES (PUSH S KEY)
LOADING END     : YES (PUSH 0 KEY)
```

For the Series 16/18–TTA onty

(5) On CNCs other than the Series 16/18-TTA, performing the operation in step (3) displays the following screen. On the Series 16/18-TTA, selecting a tool post displays the following screen and initializes the RAM.
The following screen remains displayed while the RAM is being initialized. If the RAM has low capacity, the screen in step (6) may be displayed instead of the following screen.
The selected tool post, HEAD 1 or HEAD 2, is displayed on the second line of the screen for the Series 16/18-TTA.

```
MACRO COMPILER/EXECUTOR P–CODE LOADER
HEAD 1/HEAD 2
(I/O PARAMETERS)           (DATA TRANSFER)

I/O CHANNEL   = 0          (MAPPING – 1MB)
I/O UNIT      = 0
BAUDRATE      =11
STOP BIT      = 1
INPUT CODE    = 1



OMM RAM MODULE INITIALIZE
```

Screen for the Series 16/18-TTA (The screens shown in subsequent steps are for the same CNC series.)

(6) When the RAM module has been initialized, the following loading screen is displayed:

```
MACRO COMPILER/EXECUTOR P–CODE LOADER

(I/O PARAMETERS)            (DATA TRANSFER)

I/O CHANNEL   = 0           (MAPPING – 1MB)
I/O UNIT      = 0
BAUDRATE      =11
STOP BIT      = 1
INPUT CODE    = 1



LOADING READY ? : YES (PUSH 1 KEY)
```

(7) When the P-G or PC is ready for data transmission, press numeric key 1. Then, the system waits for loading.
When this screen is displayed, execute a data transfer command on the P-G or PC to start data transmission.

```
MACRO COMPILER/EXECUTOR P–CODE LOADER

(I/O PARAMETERS)            (DATA TRANSFER)
                             ADR000000H:00
I/O CHANNEL   = 0           (MAPPING – 1MB)
I/O UNIT      = 0           ________________
BAUDRATE      =11           ________________
STOP BIT      = 1           ________________
INPUT CODE    = 1           ________________
                            ________________
                            ________________
                            ________________
                            ________________
```

(8) When the Series 16/18–A receives data, the following data reception state is displayed.
The transmitted data and its address are displayed.
Mapping is merely a term indicating how much RAM the created P-CODE program and executor occupy.
A single asterisk (∗) refers to 16K bytes when a 2MB RAM module is installed and 8K bytes when a RAM module of 1MB or less is installed.

- When a RAM module of 1MB or less is installed

```
MACRO COMPILER/EXECUTOR P–CODE LOADER

(I/O PARAMETERS)            (DATA TRANSFER)
                              ADRxxxxxxH:xx
I/O CHANNEL   = 0           (MAPPING – 1MB)
I/O UNIT      = 0           *****___________
BAUDRATE      =11           ________________
STOP BIT      = 1           ________________
INPUT CODE    = 1           ________________
                            ________________
                            ________________
                            ________________
                            ________________
```

- When a 2MB RAM module is installed

```
MACRO COMPILER/EXECUTOR P–CODE LOADER

(I/O PARAMETERS)            (DATA TRANSFER)
                              ADRxxxxxxH:xx
I/O CHANNEL   = 0           (MAPPING – 2MB)
I/O UNIT      = 0           *****___________
BAUDRATE      =11           ________________
STOP BIT      = 1           ________________
INPUT CODE    = 1           ________________
                            ________________
                            ________________
                            ________________
                            ________________
```

(9) When loading terminates normally, the following message is displayed. Pressing numeric key 1 starts the Series 16/18–A and enables the executor to execute the macros loaded in the RAM.

```
MACRO COMPILER/EXECUTOR P–CODE LOADER

(I/O PARAMETERS)             (DATA TRANSFER)
                              ADRxxxxxxH:xx
I/O CHANNEL   = 0            (MAPPING – 1MB)
I/O UNIT      = 0            ***************
BAUDRATE      =11            *****__________
STOP BIT      = 1            _______________
INPUT CODE    = 1            _______________
                             _______________
                             _______________
                             _______________
                             _______________

LOADING END   : TOTAL xxxxxxxxBYTES
 PUSH 1 KEY   : CNC START WITH executor
```

When loading terminates normally for the Series 16/18-TTA, the following message is displayed.
Pressing numeric key 1 returns to the screen in step (4). Select the next tool post on the screen.

```
MACRO COMPILER/EXECUTOR P–CODE LOADER
HEAD 1
(I/O PARAMETERS)             (DATA TRANSFER)
                              ADRxxxxxxH:xx
I/O CHANNEL   = 0            (MAPPING – 1MB)
I/O UNIT      = 0            ***************
BAUDRATE      =11            *****__________
STOP BIT      = 1            _______________
INPUT CODE    = 1            _______________
                             _______________
                             _______________
                             _______________
                             _______________

LOADING END   : TOTAL xxxxxxxxBYTES
 PUSH 1 KEY   : RETURN TO HEAD SELECTION
```

(10) If an error occurs during loading, the following message is displayed. To perform loading again, press the CAN key to retry from the beginning, that is, initializing the RAM in step (5).
If an error such as a parameter setting error occurs in the Series 16/18–A, turn the power to the Series 16/18–A off then on again.

```
MACRO COMPILER/EXECUTOR P–CODE LOADER

(I/O PARAMETERS)            (DATA TRANSFER)
                             ADRxxxxxxH:xx
I/O CHANNEL   = 0           (MAPPING – 1MB)
I/O UNIT      = 0           **********_____
BAUDRATE      =11           _______________
STOP BIT      = 1           _______________
INPUT CODE    = 1           _______________
                            _______________
(ERROR OCCURED)             _______________
ILLEGAL CHARACTER           _______________
                            _______________

LOADING RESTART ?   : YES (PUSH CAN KEY)
                    : NO  (TURN OFF POWER)
```

(11) When an executor option or reader/punch control option is not selected, the following message is displayed. In this case, loading cannot be performed. Turn off the power.

```
MACRO COMPILER/EXECUTOR P–CODE LOADER

(I/O PARAMETERS)            (DATA TRANSFER)

I/O CHANNEL   = 0           (MAPPING – 1MB)
I/O UNIT      = 0
BAUDRATE      =11
STOP BIT      = 1
INPUT CODE    = 1


executor OPTION NOTHING
I/O OPTION NOTHING
I/O OPTION (CH2) NOTHING

PLEASE TURN OFF POWER
```

## 2.1.6.2 Notes

(1) Either channel 1 or 2 is used for I/O for the Series 16 during loading. Data cannot be loaded using other channels.

(2) Either ASCII or ISO code is used to transmit data. EIA codes cannot be used.
The setting of parameter ASI (bit 3 of parameter 101, 111, or 121) determines whether ASCII or ISO code is used to transmit data.

(3) The settings of RS-232C parameters described in items (1) and (2) are displayed in the fields of I/O parameters on the loading screen. However, the parameters cannot be set on the screen.

(4) Whether P-code data is transmitted together with the executor depends on the operation of the macro compiler even when parameter PCODE for loading by overwriting (bit 1 of parameter 8701) is 1.

(5) When data is transferred for the first time after the RAM module is installed, set parameter PCODE (bit 1 of parameter 8701) to 0, so that the P-code data is transferred together with the macro executor.

### 2.1.6.3 Parameters

| No. | |
|---|---|
| 0020 | Selection of an I/O device: I/O CHANNEL |

Input setting enabled

Data type : Bytes

Valid range : 0 to 3

Selects an I/O device to be used.

0 : The device for channel 1 is selected.
(I/O device connected to JD5A of the main CPU board)

1 : The device for channel 1 is selected.
(I/O device connected to JD5A of the main CPU board)

2 : The device for channel 1 is selected.
(I/O device connected to JD5B of the main CPU board)

Either channel 1 or 2 can be used for I/O for the P-code loader.

| No. | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 0101 | | | | | ASI | | | SB2 |

Data type: Bits

SB2 : The number of stop bits is:
0 : 1
1 : 2

ASI : The following code is used for the data to be input.
0 : ISO code
1 : ASCII code

**NOTE**
The P-code loader cannot use EIA code.

| No. | |
|---|---|
| 0102 | Specification number of the I/O device (when I/O CHANNEL = 0) |

Data type: Bytes

Specify the number of the I/O device when I/O CHANNEL = 0 as follows:

**Table 1**

| Setting | I/O device |
|---|---|
| 0 | RS-232C |
| 1 | FANUC CASSETTE ADAPTOR 1 (FANUC CASSETTE B1/B2) |
| 2 | FANUC CASSETTE ADAPTOR 3 (FANUC CASSETTE F1) |
| 3 | FANUC PROGRAM FILE MATE<br>FANUC FA CARD ADAPTOR<br>FANUC FLOOPY CASSETTE ADAPTOR<br>FANUC SYSTEM P MODEL H |
| 4 | Not used |
| 5 | Portable tape reader |
| 6 | FANUC PPR<br>FANUC SYSTEM P MODEL G<br>FANUC SYSTEM P MODEL H |

Specify 0 to use the P-code loader.

| No. | |
|---|---|
| 0103 | Baud rate (When I/O CHANNEL = 0 |

Data type: Bytes

Specify the baud rate for the I/O device when I/O CHANNEL = 0 according to Table 2:

**Table 2**

| Setting | Baud rate | Setting | Baud rate | Setting | Baud rate |
|---|---|---|---|---|---|
| 1 | 50 | 5 | 200 | 9 | 2400 |
| 2 | 100 | 6 | 300 | 10 | 4800 |
| 3 | 110 | 7 | 600 | 11 | 9600 |
| 4 | 150 | 8 | 1200 | 12 | 19200 |

Specify the same baud rate for the P-code loader as that for the PC or P-G.

| No. | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 0111 | | | | | ASI | | | SB2 |

Data type: Bits

This parameter must be set when I/O CHANNEL = 1. The meanings of the bits correspond to those of the bits in parameter 101.

| No. | |
|---|---|
| 0112 | Specification number of the I/O device (when I/O CHANNEL = 1) |

Data type: Bytes

Specify the number of the I/O device when I/O CHANNEL = 1 according to Table 1.

| No. | |
|---|---|
| 0113 | Baud rate (when I/O CHANNEL = 1) |

Data type: Bytes

Specify the baud rate for the I/O device when I/O CHANNEL = 1 according to Table 2.

| No. | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 0121 | | | | | ASI | | | SB2 |

Data type: Bits

This parameter must be set when I/O CHANNEL = 2. The meanings of the bits correspond to those of the bits in parameter 101.

| No. | |
|---|---|
| 0122 | Number of the I/O device (when I/O CHANNEL = 2) |

Data type: Bytes

Specify the number of the I/O device when I/O CHANNEL = 2 according to Table 1.

| No. | |
|---|---|
| 0123 | Baud rate (when I/O CHANNEL = 2) |

Data type: Bytes

Specify the baud rate for the I/O device when I/O CHANNEL = 2 according to Table 2.

| No. | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 8701 | | | | | | | PLD | |

Data type: Bits

PLD In the P-code loader for the macro compiler or executor, the RAM is:

0 : Cleared with OMM and rewritten.
1 : Not cleared with OMM, but overwritten.

## 2.1.6.4 Alarms

| Message | Explanation |
|---|---|
| EXECUTOR OPTION NOTHING | A macro executor option is not yet selected. |
| I/O OPTION NOTHING | Reader/punch control 1 option is not yet selected. |
| I/O OPTION (CH2) NOTHING | Reader/punch control 2 option is not yet selected. |
| ILLEGAL CHANNEL | A channel other than channel 1 or 2 is selected. |
| ILLEGAL CHARACTER | The transmitted data is not correct. |
| OVERRUN | Overrun error |
| FRAMING ERROR | Framing error |
| DSR SIGNAL OFF | Communication error |
| PARITY ERROR | Data without parity was transmitted. |
| CHECK SUM ERROR | Checksum error |

## 2.2 MACRO EXECUTOR CONTROLS

The macro executor controls the execution of the P-CODE program created by the macro compiler.

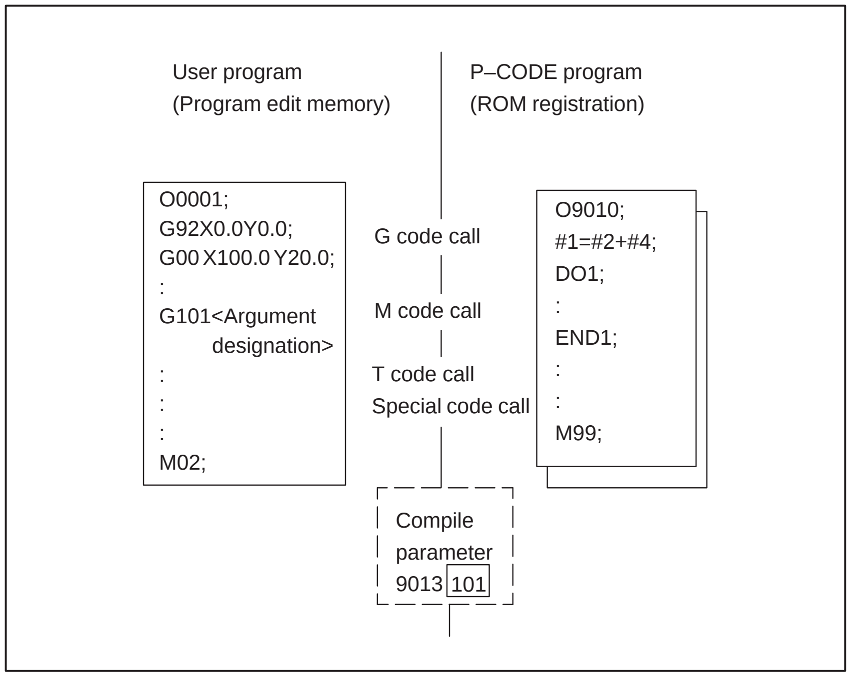
The P-CODE program (execution macro) stored in the ROM can be called and executed by specifying G, M, T or original code set by parameter during compilation in the user program.

Custom macros can be created and executed in the user program, independently of P-CODE program.

# 3 EXECUTION MACRO

Only a registered P-CODE program cannot be executed. It is called from the user program by G, M, T code, or specified code by parameter setting, and executed. In case of macro call, argument designation is possible, and it is compared as a local variable at the P-CODE (execution macro) side.

Moreover, if a minus value is set to a parameter for macro call by G code, modal call of P-CODE program can be done by corresponding G code. Refer to 3.1.5 for details.



## 3.1 CALL CODE AND PROGRAM NO.

To call a P-CODE program from the user program, the codes shown in Table 3.1 are used.

**Table 3.1 Codes for Calling P–CODE program**

| Codes for call | Type of call | Program numbers called | Common variables in which a specified code is stored | Parameters to be set |
|---|---|---|---|---|
| T | Subprogram call | 9000 | #149 | Compile parameter TCAL(No.9002#0) |
| M | Subprogram call | 9000-9003 | None | Compile parameters No.9010-9012 |
| Specified code | Subprogram call | 9004<br>9005 | #146,#147 | Compile parameter ACL1,ACL2<br>(No.9002#1,#2) |
| G | Macro call,<br>Modal call | 9010-9019 | None | Compile parameters No.9013-9022, No.9034 |
| M | Macro call | 9020-9029 | None | Compile parameters No.9023-9032 |
| T | Macro call | 9008 | #27 | Compile parameter TMACC(No.9005#7) |
| M (range specification) | Sbprogram call | 9009 | #148 | Compile parameters No.9042,9043 |
| G (range specification) | Macro call | Parameter | None | Compile parameters No.9045-9047 |
| Address for axis | Macro call | 9009 or 9031 to 9038 | #27 | Compile parameter<br>AX*CL(No.9005#0-#3, No.9008#0-#3)<br>AXCLS (No.9005#4) |

**NOTE**
These codes cannot be used in combination. For example, a macro call by a T code cannot be executed together with a subprogram call by an M code.

Correspondence between codes that call macro programs or subprograms and program numbers of called programs, and whether or not to call a subprogram or a macro , are determined by compile parameters. Since these parameters are registered to the ROM at compilation, be sure to designate them at compilation. Exclusive codes that call subprograms shall be set to CNC parameter (No,6090,6091) in executing.

(1) The return sequence number definition for returning to the user's program
When operational control is returned to the user's program from the P-CODE program, control passes to the sequence number of the user's program defined by address P.

| User's program | | P–CODE program |
|---|---|---|
| O0001; | | |
| : | | |
| G100X123Y456; | → | O9010; |
| G00X789; | | #100=#24+#25; |
| : | | : |
| N100G00Y999; | ← | M99P100; |
| : | | |
| M02; | | |

(2) Difference between Subprogram Call (T, M) and Macro Call (G, M)

(a) Argument designation can be made in macro call. In subprogram call, however, argument designation is not possible without T code, special code and call code.

(b) In subprogram call, after execution of another command than T- or M-code, it will branch to a subprogram. In macro call, however, it will branch off without doing anything.

(c) In subprogram call, single block stop is made when another command than T- or M-code is commanded. In macro call, however, no stop is made.

## 3.1.1 Calling Subprogram O9000 by T code

The P-CODE program 09000 registered to the ROM can be called by a T code.

The commanded T code is stored as an argument in the common variable #149.

All the local variables will become <Vacant>.

N_ G_ X_ Y _ T < tttt >;

## 3.1.2 Calling subprograms O9001–9003 by M code

By commanding M codes which is designated by compilation parameter, the programs O9001, O9002, O9003" registered to the ROM can be called for subprograms. All the local variables are <Blank>.

N_ G_ X_ Y _ M<mm> ;

## 3.1.3 Calling a Subprogram Using Specified Codes

By setting character codes (decimal notation of ASCII codes) to the parameters (No. 6090, 6091) of CNC, the P-CODE program (09004, 09005) corresponding to the address can be called as a subprogram. The defined integer will be stored as a parameter to the macro variable (#146, #147). The actual use of this function is decided by specifying the appropiate compilation parameter (No. 9002#1, 9002#2, ACL1, ACL2).

Example) Compiler parameter (No. 9002#1, ACL=1)
When parameter No. 6090 = 66 at execution

| User's program | | P–CODE program |
|---|---|---|
| O0001; | | |
| O0001; | | |
| : | → | O9004;(#145=100.) |
| G00X123Y456B100; | | |
| : | | |
| M02; | | |

Definable addresses for 16-TB/16-TA/GCA, 18-TB/18-TA/GCA, 20-TA, 21-TB: A, B, D, F, H, I, J, K, M, Q, R, S, T

Definable addresses for 16-MB/16-MA/GSA, 18-MB/18-MA/GSA, 20-FA, 21-MB: A, B, F, H, I, K, M, Q, R, S, T

## 3.1.4 Calling Macros O9010–9019 by G Code

By commanding G codes which is designated by compilation parameter (No. 9013 to 9022), the macro programs ”O9010-O9019” registered to the ROM can be called.

Local variables without argument designation are <Vacant>.

N_ G<gg>< Argument designation>;

## 3.1.5 Modal Call Using G Code

A modal call can be used to call a macro when using G code.

When calling a G code is specified with a compilation parameter, continuous-state calling can be specified using a negative number.

Continuous-state calling is canceled with G167 or the G code specified by compilation parameter 9034.

During continuous-state calling, the values of address of each block for the user program are all arguments.

Example) When compilation parameter 9013 = –100

| User's program | P–CODE program |
|---|---|
| O0001; | |
| : | |
| G100X123Y456; | O9010 |
| X789 Z678; | for each block the difined address is |
| : | called as a parameter of “09010” |
| : | and executed. |
| G167; | |
| (Modal call cancelled or code set to compile parameter No.9034.) | |
| : | |
| M02; | |

Multiple modal calls is not permitted.

## 3.1.6 Calling Macros O9020–9029 by M Code

By commanding M codes which is designated by compilation parameter (No. 9023 to 9032). the programs ”O9020 - O9029” registered to the ROM can be called.

Local variables without argument designation will become <Vacant> .

N_ M<mm>< Argument designation>;

## 3.1.7 Calling a Macro Using a T Code

By setting compile parameter TMACC(No.9005#7) to 1, program of No.9008 registered to ROM can be called macros by specifying a T code in a user program.

All addresses specified in this block are used as arguments except that the T code is transferred to #27, values for address P and L are transferred to #16 and #12, respectively. Also G codes are transferred to variables #28 to #32 for each group.

Be sure that addresses shall be those availabe for CNC and the significant digits are those specified by the CNC.

| Variable | Data to be transferred |
|---|---|
| #1-#26 | Address data for each variable |
| #27 | T code |
| #28 | G code |
| #29 | G code |
| #30 | G code |
| #31 | G code |
| #32 | G code |

**NOTE**
G codes are assigned to variables #28 to #32 in ascending order, starting with the lowest G code group number. When a G code of G code group 01 exists, G80 may be generated and assigned to #28 to #32.

Example) G91G28X123.45678T5678:
#24=123.456
#27=5678.0
#28=28.0
#29=91.0
Other variables = < vacant >

## 3.1.8 M Code Subprogram Call with Range Specified

ROM-resident program 09009 can be called as a subprogram by using compilation parameters No. 9042 and No. 9043.

Specify the following codes in compilation parameters No. 9042 and No. 9043:

No. 9042 = M code indicating the lower limit

No. 9043 = M code indicating the upper limit

Examples are given below:
No. 9042 = 100
No. 9043 = 110

If the above codes are specified in the parameters upon compilation, specifying an M code within the range from M100 to M110 calls ROM-resident program O9009 as a subprogram.

## 3.1.9 Calling Macros with a G Code by Specifying the Range

The code specified by compilation parameters 9045 to 9047 can be used to call the programs registered in ROM.

Specifying call arguments is the same as calling macros with G code (3.1.4).

Specify the following for parameters 9045 to 9047:

Parameter 9045 : G code to start calling

Parameter 9046 : Number of P-CODE programs

Parameter 9047 : Number of the program to be called first

For example, suppose programs are compiled with 200 specified for parameter 9045, 100 specified for parameter 9046, and 1000 specified for parameter 9047. When G200 to G299 are specified, 100 programs from O1000 to O1099 compiled on the ROM can be called.

**NOTE**

1 Specifying call arguments is the same as calling macros with the G code specified by compilation parameters 9013 to 9022.

2 Continuous-state calling cannot be specified.

3 When the G code specified with compilation parameters 9013 to 9022 is specified, the specified parameters 9013 to 9022 are effective.

Parameter 9013 = 250
Parameter 9045 = 200
Parameter 9046 = 100
Parameter 9047 = 1000

When G250 is issued with the settings above, program O9010 is called.

## 3.1.10 Function for Calling Macros with an Axis Address

Axis address commands enable calling macros.

When AX1CL to AC8CL of compilation parameter 9005#0 to #3, 9008#0 to #3 are set to 1, the programs registered to ROM can be called by the axis address command.

The program number to be called is selected by compile parameter AXCLS (No. 9005#4) as follows:

AXCLS 1 : The program number to be called depends on a specified axis:
Program O9031 is called when 1st axis is specified.
Program O9032 is called when 2nd axis is specified.
: :
Program O9038 is called when 8th axis is specified.

0: Always program O9009 is called irrespective of specified axes.

In this case, all the addresses in the block specified axis address are passed for use as arguments. However, the specified axis address is passed to variable 27. Addresses P and L are passed to variable 16 and 12, respectively, for use as arguments. Up to five G codes in each G code group are passed to variables 28 to 32 starting from the group with the lowest number. When a G code of G code group 01 exists, G80 may be generated and assigned to #28 to #32.

When G code system A is used in the lathe system (T/TT), whether the axis addresses are called with an absolute or incremental command are informed to variable 33.

| Variable No. | Address |
|---|---|
| #1 to #26 | Usual argument address |
| #27 | Specified axis address (1st to 8th) |
| #28 | Specified G code |
| #29 | Specified G code |
| #30 | Specified G code |
| #31 | Specified G code |
| #32 | Specified G code |
| #33 | Vacant for absolute address<br>1.0 for incremental address |

If the following are specified when a 4th-axis address is B, for example:
G91G28B1.234567X123.4567;

The settings are passed to variables as follows:
123.456 to variable #24
1.234 to variable #27
28.0 to variable #28
91.0 to variable #29
Other variables: Null

The addresses transferred as arguments specified in a block used to call a macro make changes modal information of the CNC when calling the macro. In the above example, the CNC model of absolute/incremental command changes to G91 (incremental command) when a macro is called.

In the lathe system, for G code system A, and when a 2nd-axis address is Z (absolute) or W (incremental), the settings are passed to variables as follows:

Variable #27 = 100.0, variable #33 = <null> for Z100.;

Variable #27 = 100.0, variable #33 = 1.0 for W100.;

**NOTE**
The addresses those can be used and the range of the values specified to those addresses are the same as those allowed to each CNC model concerned.

## 3.2 ARGUMENT DESIGNATION

Argument designation is possible when calling a call. It can be referred to as a local variable at the P-CODE program side. Argument specification I and argument specification II are possible.

For arguments designation, negative symbol and decimal point can be used irrespective of the address.

**Table 3.2(a) Argument specification I at P–CODE program call**

| Address of argument specification I | Local variable No. | Address of argument specification I | Local variable No. |
|---|---|---|---|
| A | #1 | N (NOTE) | #14 |
| B | #2 | P (NOTE) | #16 |
| C | #3 | Q | #17 |
| I | #4 | R | #18 |
| J | #5 | S | #19 |
| K | #6 | T | #20 |
| D | #7 | U | #21 |
| E | #8 | V | #22 |
| F | #9 | W | #23 |
| G (NOTE) | #10 | X | #24 |
| H | #11 | Y | #25 |
| L (NOTE) | #12 | Z | #26 |
| M | #13 | | |

**NOTE**
It is available when compilation parameter bit5 of No.9008 is 1.

**Table 3.2(b) Argument specification II at P-CODE program call**

| Address of argument specification II | Local variable No. | Address of argument specification II | Local variable No. |
|---|---|---|---|
| A | #1 | K1 | #6 |
| B | #2 | .<br>. | .<br>. |
| C | #3 | I10 | #31 |
| I1 | #4 | J10 | #32 |
| J1 | #5 | K10 | #33 |

## 3.3 LIMITATION FOR EXECUTION MACRO

Source program of execution macro should be programmed in the same way as for custom macro. But, there are some limits for execution.

(1) Macro call

Macro call from an execution macro is executed with ”G65” as the custom macro . In the execution macro , since it is a macro program itself to be called from the user program with G (M, T) code or specified code, it is impossible to use a G CODE call, etc. from execution macro.

G65 P (Program No.) L (Number of repetition) <Argument specification> ;

(2) Argument specification

Argument specification is the same as the custom macro.

(3) Variable

Expression, argument, etc. of variables are the same as those of the Custom macro.

> **NOTE**
> Common variables referred by a P–CODE program and Common variables referred by user program are completely different.
> Refer to ”5.3 Common variable”.

(a) P–CODE variable (#10000 – )

Any number of 100 unit of P-CODE exclusive variables starting from #10000 can be used.
Since it can be used from execution, it is considered as extension of common variable #500.
However, execution macro cannot refer to P-CODE variable in arrangement type.
For details, refer to ”5.4 P-CODE exclusive variable”.

(4) Operation command

Operation commands can be used as with the custom macro.

(5) Control command

Both divergence and repetition commands can be used .

IF <Conditional expression> GOTO n;

WHILE <Conditional expression> DO m;

END m;(m=1,2,3)

(6) Modal call from execution macro

Modal call cannot be made.

(7) Macro and subprogram multiplexity in execution macro.

Separately from the user program multiplexity, 4-stack nesting of macro program, and 4-stack nesting of subprogram are possible on the execution macro.

(8) Cautions

(a) Separate registration of a program cannot be made. Max. 400 programs can be registered to the ROM.

(b) In one program, limit the sequence number used for branch destination (GOTO) to 200. In one program, the same sequence No. cannot be designated for others.

(c) One block can accept designation of only one sequence number. Except the program No. ”0” and the optional block skip ”/”, designate a sequence No. at the block head.

(d) In T series multiple repetitive canned cycle cannot be executed in the P-CODE program. In case a program including a multiple repetitive canned cycle is registered and executed, the function cannot be guaranteed.

(e) In T series programming through direct drawing dimensions programming cannot be made in the P-CODE program.
In case a program made through direct drawing dimension input is registered and executed the function cannot be guaranteed.

# 4 CONVERSATIONAL MACRO FUNCTION AND AUXILIARY MACRO FUNCTION

## 4.1 CONVERSATIONAL MACRO

The conversational macro function executes a program compiled by the macro compiler independently of the normal NC part program. This function basically is not affected by operation modes of NC . Accordingly, it works concurrently with the NC part program under execution independently even during the memory operation.

**NOTE**
The execution of the conversational macro function is processed at a lower level than that of the CNC operation internally. Therefore, the execution of the conversational macro function will not affect the processing speed of the CNC operation, but the processing speed of the conversational macro function may become slow while the CNC operation is ON.

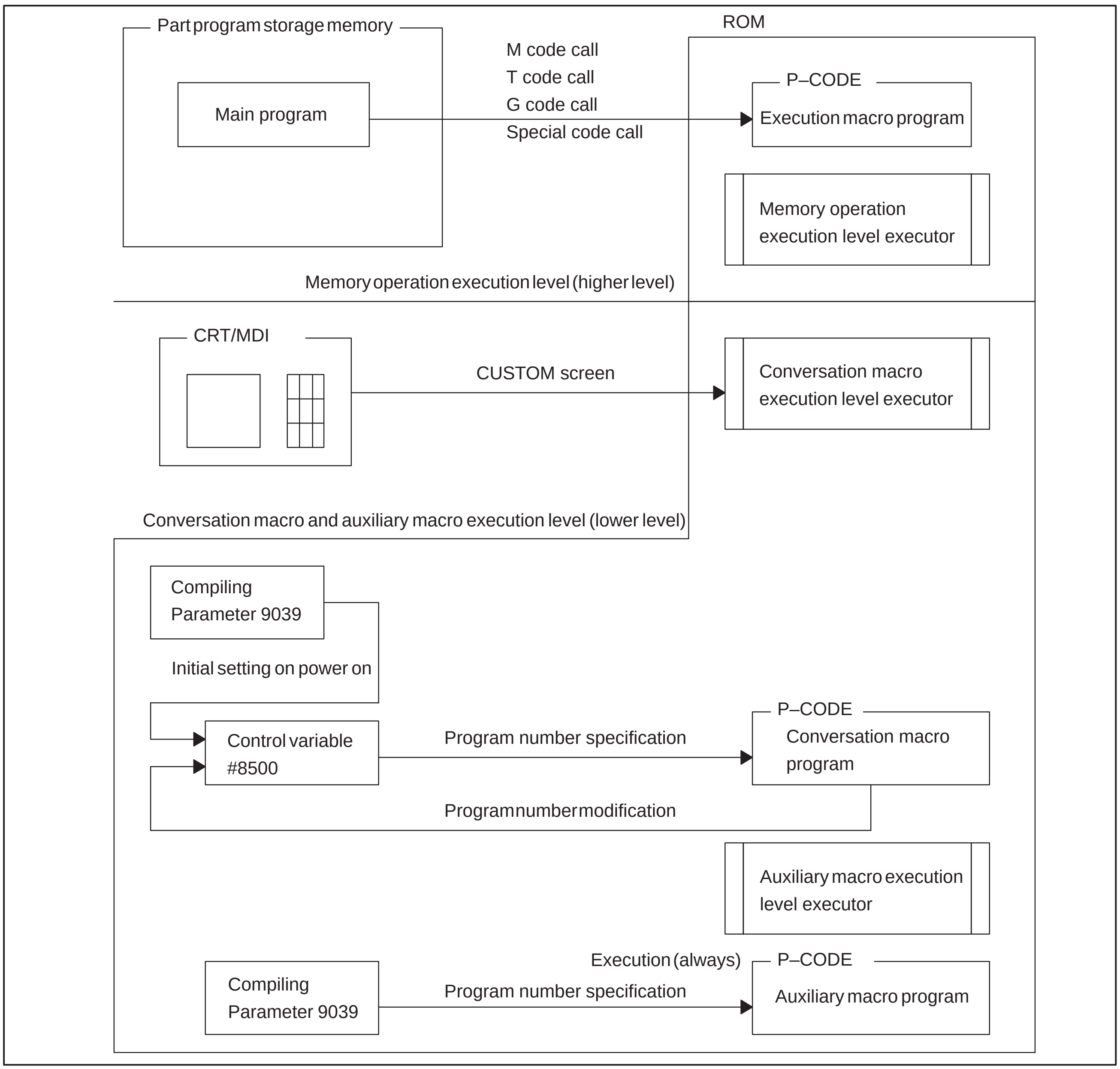



**Fig.4.1 Conversational macro function conceptional diagram**

The conversational macro function is executed only when the conversational screen is selected with the function key CUSTOM .

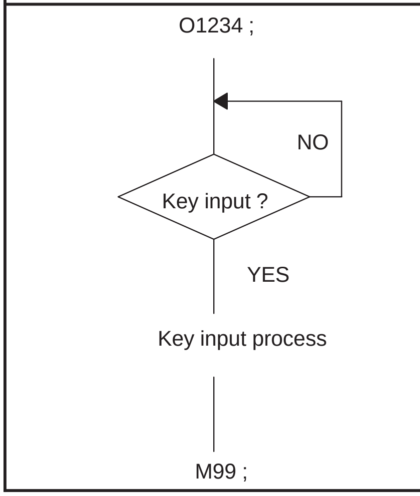
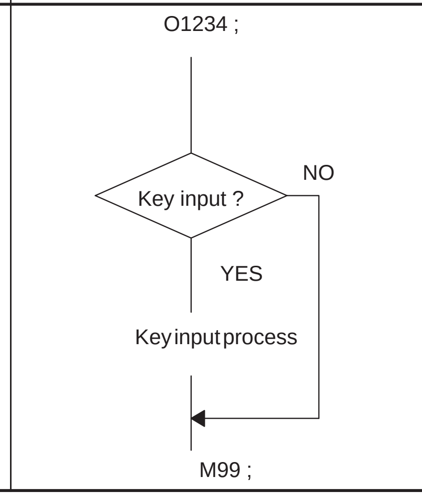
The conversational macro function executes its macro program whose program No. is the value of the conversational macro execution control variable #8500. The value of the conversational macro execution control variable #8500 is to be set with the parameter No. 9038 at compiling. The conversational macro function will not be executed when there is not a conversational macro program compiled whose program No. is the value of the conversational ma ro execution control variable #8500.

The conversational macro program, as well as NC part program, consists of macro sentences and NC sentences (specific G code commands).

The conversational macro program can describe all the macro statements used for the custom macro. And all the system variables and common variables which can be referred to in the P-CODE program can be read and written. But the conversational macro program has no local variables except for #1 to #99. They are used to refer an array type P-CODE variables #10000 's. Also, P-CODE variables are prepared for the conversational macro control.

It is impossible to execute the NC part program statement with the NC statement of the conversational macro program. They will be ignored though commnded. It is impossible to use any codes except the screen display codes and the execution control codes described in the following sections. And be sure that the NC statement of the conversational macro program have the addresses whose meanings and usage are different from those of the NC part program statement.

The CUSTOM screen started by the conversational macro program is under the same control as the other screens (POS screen, etc.). Therefore, finishing of the CUSTOM screen is necessary for changing over from one screen to another. Decide the timing to finish when executing M99 of the main program. Be careful about the condition of the conversational macro program, because if it is of a bad program example as shown below, it will be impossible to change the screen to another and HANG-UP STATE will be held. When another function key is pressed mistake, press CUSTOM again.

| Bad example | Good example |
|---|---|
|   |   |

Program the conversational program so that it will be a cyclic like the PMC ladder program. That is, execute M99 without fail and it will return to the head of the main program or to the sequence No. specified with M99Pp.

Thus, avoid the programming that the divergence destination by GOTO will reverse direction.

**NOTE**

To select a conversational macro screen, press function key CUSTOM .

To change the screen from the conversational macro screen to another screen such as current position screen or program screen, press the correspondence function key such as POS and PROG .

If the Series 20 is your NC, press both right most and left most soft keys at the same time and the screen changes from the conversational macro screen to the current position display screen.

## 4.2 AUXILIARY MACRO FUNCTION

The auxiliary macro is always executed regardless of the selection screen while the conversational macro function is executed only when the conversational macro screen is selected in the function CUSTOM .

The auxiliary macro function executes the auxiliary macro program with the program number set by parameter 9030 on compiling.

If parameter 9039 is equal to 0 or the auxiliary macro with a set program number has not been compiled, no auxiliary macro function is executed.

Major differences between the conversation macro and auxiliary macro functions are as in the following:

1) Auxiliary macro function is always executed regardless of the screen being displayed.
2) No screen display control code can be used in the auxiliary macro function. G202, G240, G242, G243, G01, G02, G03
3) No variables for controlling and cursor can be used in the auxiliary macro function.
#8501-#8509
4) The conversational macro function can control the program number executed by conversational macro execution control variable #8500. On the other hand, the auxiliary macro function always executes the program of the program number set by parameter 9039 on compiling.

**NOTE**
1 Switching to another screen is disabled while an auxiliary macro is controlling the reader/punch interface.
2 A larger auxiliary macro (requiring a longer time to execute) causes screen drawing to become slower.

The conversation macro function and auxiliary macro function are executed at the same execution level (lower level).
The execution processing is as shown in the following when both the conversation macro function and auxiliary macro function are specified.

1) CUSTOM screen selected
The auxiliary macro execution and conversation macro executed are repeated in this order.
2) Screen other than CUSTOM screen
A compilation parameter makes it possible to display CUSTOM screen at power on.

## 4.3 EXECUTION CONTROL CODE

The following control codes are prepared for execution control.

M98 : Subprogram call

M99 : Subprogram end

(1) Subprogram call (M98)

M98 Pp ;

A macro subprogram specified by address P is called. Frequency of calling the macro subprogram shall be up to quadruple.

(2) Subprogram end (M99)

M99 Pp ;

When command is done in the c lled subprogram, the calling program will be restored. When the address P is specified, the block of the sequence No. specified at P of the calling program will be restored.

Command M99 without fail at the end of the main program. The command of M99 in the main program is explained below.

When M99 is commanded in the main program, execution of the conversational macro function is finished once.

Once execution of the conversational macro function is started, the screen will not change over to another though the function key is depressed until execution of the conversational macro function ends at M99. Accordingly, it is necessary to command M99 at the end of the main program.

When the function key is depressed during the execution of the conversational macro program, both character display and graphic display will be erased after the conversational macro program ends, and a corresponding screen will take place. When the function key CUSTOM is depressed again, the execution of the conversational macro program will be started according to the value of the conversational macro execution control variable #8500. At this time, the program will be executed from the beginning regardless of the command of the address P in block of M99.

When M99 is executed, the value of the conversational macro execution control variable #8500 will be checked. When the value of #8500 is rewritten, both character display and graphic display will be erased and the control will be transferred to a new conversational macro program. If not, neither the character display nor the graphic display will be erased, and execution of the same conversational macro program will be repeated. In this case, when the address P is commanded, execution will be started at the block of the sequence No. specified at P.

Once M99 is executed, the graphic display will not be output on th screen until the graphic screen is erased next. Consequently, when the same conversational macro program is still executed, the second and the following graphic displays will not be output on the screen When you want to redisplay the graphic display in 1 conversational macro program, command the graphic screen to be erased once.

## 4.4 CONVERSATIONAL MACRO EXECUTION CONTROL VARIABLE #8500, #8550, #8551

Three screens are available for execution of the conversational macro. These screens are selected by pressing the function key CUSTOM. The three screens correspond with those for the FANUC Series 0 as follows;

| Screen for Series 16/18/20 | Screen for Series 0 | Conversational macro execution control variable |
|---|---|---|
| CUSTOM screen 1 | AUX | #8500 |
| CUSTOM screen 2 | MENU | #8550 |
| CUSTOM screen 3 | MACRO | #8551 |

When the function key CUSTOM is pressed, a conversational macro whose program number is specified by a value of the conversational macro execution control variale is executed.

Compile parameters (No.9038,No.9040,No.9041) set values of the conversational macro execution control variable effected at power on. When the value of the conversational macro execution control variable is re-written, another conversational macro is controlled. When the value is re-written, characters and graphics are erased after completion of M99 of the currently executed conversational macro program(main) and the new conversational macro is controlled.

If the value of the conversational macro execution control variable is not re-written, the same conversational macros are repeatedly executed. In this case, no characters and graphics are erased.

## 4.5 NOTES

The conversational macro and auxiliary macro must be programmed so that M99 of the main program is executed.

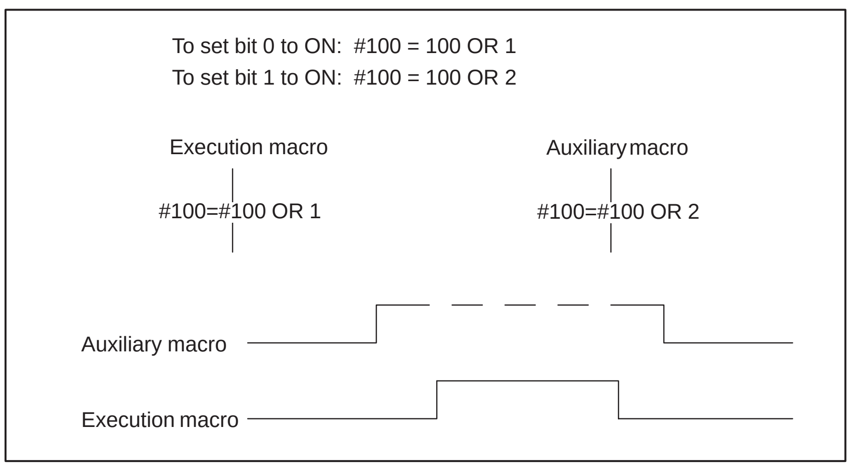
The conversational macro, auxiliary macro and standard CNC display (POS screen, etc.) are executed on the same level. If a wait is programmed in any macro not having M99, the following occurs:

- If the wait is executed by the conversational macro, the auxiliary macro is not executed.
- If the wait is executed by the auxiliary macro, the conversational macro is not executed. The CNC display (POS screen, etc.) is not updated.

When common variables are used by an auxiliary macro (or conversational macro) and execution macro, the same variable must not be written to as a flag. An execution macro has higher priority than an auxiliary macro (or conversational macro). So, while an auxiliary macro (conversational macro) is writing to a variable, an execution macro may interrupt and write to that variable.

In such a case, once writing to the variable by the execution macro ends, the remaining write processing by the auxiliary macro (or conversational macro) is completed. So, the value written by the execution macro may be overwritten by the auxiliary macro (or conversational macro).

Example: Bit 0 of the #100 value is used as an execution macro flag, while bit 1 of #100 is used as an auxiliary macro flag




The value of #100, read at the start of auxiliary macro processing, is written to #100 after auxiliary macro processing ends. So, a value written by execution macro processing may be lost.

# 5 MACRO VARIABLES

## 5.1 MACRO VARIABLES

| Variable | No. | User program | Conversational macro | Auxiliary macro | Executable macro |
|---|---|---|---|---|---|
| Local variable | #1 to #33 (#99) | ○ | Δ (NOTE 1) | | ○ |
| Common variable | #100 and up<br>#500 and up | ○ | ○ (Common variable) | | |
| P-CODE variable<br>Extended P-CODE variable | #10000 and up<br>#20000 and up | × | ○ (Common variable) | | |
| Control variable | #8500 and up | × | ○ | Δ (NOTE 2) | × |

○ : Usable Δ : Partially usable × : Unusable

**NOTE**
1 To be used when the P-CODE variables of array type are referenced.
2 The variables for controlling screen display and key input cannot be used.

## 5.2 LOCAL VARIABLES (#1 TO #33 OR FOR REFERENCING THE P–CODE VARIABLES OF ARRAY TYPE, #1 TO #99)

The local variables can be used for executable macros.

These local variables are different from those used for the user programs.

The local variables can be used for auxiliary and conversational macros to reference the P-CODE variables of array type.

## 5.3 COMMON VARIABLES (#100 TO #149 AND #500 TO #531)

The common variables can be used for conversational macros, auxiliary macros, and executable macros.

The common variables are common to the conversational, auxiliary, and executable macros.

However, they are different from those used for the user programs.

## 5.4 P–CODE VARIABLES #10000–

It is possible to use optional quantity of the P-CODE variables starting with #10000 on 100-pc. basis.

A 100 times as many as the numerical value preset on the compiling parameter No. 9037 will be the usable quantity of P-CODE variables. When the value on the parameter No. 9037 is 0, P-CODE exclusive variables are not usable.
The upper limit of the P-CODE variables is as follows.
10000 + (Value of Compiling Parameter No. 9037) $\times$100–1

The lower limit of the P-CODE variables is 10000. For example, the P-CODE variables can be used as follows:
#10000 to #10099 when the value of parameter No. 9037 is 1
#10000 to #10199 when the value of Parameter No. 9037 is 2

**NOTE**

1 For P-CODE variables, about 1.63 meter of part program memory is consumed per 100 pieces of variables when part program storage is 20 to 80m. The more P–CODE are used, the more part program memory decreases. The maximum no. of the usable P-CODE variables depends on the capacity of the part program memory.
Part program storage memory 10 m:
Parameter on compiling No. 9037 = 6
Part program storage memory 20 m:
Parameter on compiling No. 9037 = 12
Part program storage memory 40 m:
Parameter on compiling No. 9037 = 25
Part program storage memory 80 m:
Parameter on compiling No. 9037 = 51
Part program storage memory 160 m - 1280 m:
Parameter on compiling No. 9037 = 60
- In Series 16-B/18-B, part program storage memory is not used. (Parameter no. 9037 needs to be set).
When compilation parameter no. 9007#7 EXT=1, the no. of P-CODE variables is the above value–1.
- For the 21-TB, approximately 1.63 meters of tape is required to store each 100 P-CODE variables, even when the tape length is 160 meters.
- For the 21-MB, whether part program memory is used varies with the CNC system software series.
  - DDA1 series : Part program memory is not used.
  - D201 series : Part program memory is used.

2 The P–CODE variables still maintain their values after the power is turned off.

3 Before executing the conversational macro with the complied ROM module loaded, erase all the programs in the part program storage memory once.

(Turn on the power while pushing DELETE key with the setting (PWE = 1).)
With this operation all the conversational macro functions will be initialized to be ”blank”.
Array type variables for P–CODE in conversational macro #10000 – can be refered to in 2 to 3 dimentional array type. (See 6.12 (1) )

## 5.5 VARIABLES OF EXPANDED P–CODE (#20000 – ....)

The variables of the expanded P-CODE which start from #20000 can be used for the optional number, if the capacity of tape memory is 160m to 1280m. This variable can be selected for numbers with floating decimal points, the same as the usual common variable or for numbers with the integer type by parameter (parameter No. 9002#3 EVF) specification.

Parameter No. 9002#3

EVF= 0: floating decimal point type

EVF= 1: the integer type

N times the number set paramater No.9044 is the number of the usuable variables for the P-CODE the same as the exclusive variables for the P-CODE (#10000 ---). The P-CODE variables cannnot be used, if parameter No. 9044 is 0.

The number of n is 12 with the floating decimal point type, and is 30 with the integer type.

EVF=0: in the case of the floating decimal point type

#20000 to #20011 if parameter No. 9044 equals 1

#20000 to #20023 if parameter No. 9044 equals 2

EVF=1: in the case of the integer type

#20000 to #20029 if parameter No.9044 equals 1

#20000 to #20059 if parameter No.9044 equals 2

In the case of the integer type, one of -32768 to 32767 can be set. Digits under the decimal point are rounded off when a value is substituted to the left side of the substitution statement.

Furthermore, this variables is evaluated after converted into the floatingdecimal point type, if this varables occurs in expression.

About a 0.21 meter of part program memory is used per set (number of parameter No. 9044) of the variables of the expanded P-CODE.

The part program memory being used is displayed on the program library screen. The maximum number of P-CODE variables depends on the capacity of the part program memory.

Part program memory 160m: Parameter No. 9044=819

Part program memory 320m: Parameter No. 9044=1638

Part program memory 640m (EVF=0): Parameter No. 9044=3276

Part program memory 640m (EVF=1): Parameter No. 9044=2184

Part program memory 1280m (EVF=0): Parameter No. 9044=5461

Part program memory 1280m (EVF=1): Parameter No. 9044=2184

**NOTE**

For Series 20, Series 21, expanded P-CODE (#2000 -) cannot be used.

## 5.6 DISPLAYING VARIABLES

The variables used for P-CODE programs, such as local variables, common variables, P-CODE variables, and extended P-CODE variables, can be displayed by pressing the function key OFFSET SETTING several times.

The values of these variables can be specified using the MDI keys.

```
P–CODE VARIABLE

    No.      DATA            No.      DATA
    0      123.45678         0      123.45678
    1      123.45678         1      123.45678
    2      123.45678         2      123.45678
    3      123.45678         3      123.45678
    4      123.45678         4      123.45678
    5      123.45678         5      123.45678
    6      123.45678         6      123.45678
    7      123.45678         7      123.45678
    8      123.45678         8      123.45678
    9      123.45678         9      123.45678

    NUM.   ________
  [      ] [      ] [      ] [      ] [NO–SEL]
```

The desired screen can be selected using page keys, cursor key, and [NO-SEL] key.

Pressing the ↑ page key returns the current screen to the previous screen. Pressing the ↓ page key displays the next screen.

A desired variable can be displayed with the numeric key and [NO-SEL] key.

The desired value can be entered for the variable at the cursor position using the numeric keys and the INPUT key. However, no values can be entered in local variables or write-inhibited system variables.

**NOTE**

1 The setting of NDP, bit 1 in parameter 9000, for the executor depends on whether variables to be used for P-CODE programs are displayed. To display the variables, set bit 1 of parameter 9000 to 1.

2 The specified values of the variables are displayed when the screen is displayed. In other words, if the values of the variables are changed while the screen is displayed, the changed values are not displayed.

3 To change the screen from the P-CODE VARIABLE screen to another screen such as current position screen or conversational macro screen, press the correspondence function key such as

POS and CUSTOM .

If the 20-FA or 20-TA is your NC, press both right most and left most soft keys at the same time and the screen changes from the P-CODE screen to the current position display screen.

# 6 FUNCTIONS OF THE MACRO EXECUTOR

**Table 6 (a) Functions (1/2)**

| No. | Function | Outline | Conversa-tional | Auxiliary | Execut-able |
|---|---|---|---|---|---|
| 6.1 | Screen display function | Controls the display of the conversational macro screen. | | | |
| | 1 Screen display control | | ○ | × | × |
| | 2 Function screen control | | ○ | ○ | × |
| | 3 Masking conversational macro screen status display | | ○ | × | × |
| | 4 Shift for adjusting the graphic screen | | ○ (NOTE 1) | × (NOTE 1) | × |
| 6.2 | Address function | Reads data at PMC addresses. | ○ | ○ | ○ |
| 6.3 | Writing at PMC addresses | Writes data at PMC addresses. | ○ | ○ | ○ (NOTE 2) |
| 6.4 | Reader/punch interface control | Controls the RS-232C interface. | ○ | ○ | × |
| 6.5 | Reading and writing an NC program | Reads NC programs from and writes them onto the CNC tape. | ○ | ○ | × |
| 6.6 | Continuous input with the cursor and page keys | Continuously inputs data with the cursor and page keys. | ○ | × | × |
| 6.7 | Masking the display of O and N numbers | Clears the display of O and N numbers on the CUSTOM screen. | ○ | × | × |
| 6.8 | Reading and presetting the cutting time and cutting distance | Reads and presets the cutting time and the cutting distance. | ○ | ○ | × |
| 6.9 | Reading and presetting the relative coordinates | Reads and presets the relative coordinates. | ○ | ○ | × |
| 6.10 | Key/data input control | Reads the states of MDI keys and the values of entered data. | ○ | × | × |
| 6.11 | Cursor control | Displays the cursor at any position on the conversational macro display screen. | ○ | ○ | × |
| 6.12 | Processing of P-CODE variables of array type | References P-CODE variables of two- or three-dimensional array type. | ○ | ○ | × |
| 6.13 | Torque limit control | Specifies the override values of the torque limits for each servo axis. | ○ | ○ | ○ |
| 6.14 | Reading A/D converter data | Reads A/D converter data. | ○ (NOTE 3) | ○ (NOTE 3) | × |
| 6.15 | Key-in line function | Specifies a desired display position and color of the key-in line for conversational macros. | ○ | × | × |
| 6.16 | Reading the status indicating editing in the background | Reads the status indicating whether editing is currently performed in the background. | ○ | ○ | × |
| 6.17 | Reading the number of registered programs | Reads the number of the programs registered in CNC program memory. | ○ | ○ | × |
| 6.18 | Reading the unused capacity of CNC program memory | Reads the capacity of unused areas in CNC program memory. | ○ | ○ | × |
| 6.19 | Reading the remaining travel distance | Reads the remaining travel distances for each servo axis. | ○ | ○ | × |

○ : Usable  Δ : Partially usable  × : Unusable

**Table 6 (a) Functions (2/2)**

| No. | Function | Outline | Conversa-tional | Auxiliary | Execut-able |
|---|---|---|---|---|---|
| 6.20 | Use of offset memory and extended system variables in the workpiece coordinate system | Reference data in offset memory and the values of extended system variables in the workpiece coordinate system using macro variables from 100000 to 199999. | ○ (NOTE 3) | ○ (NOTE 3) | ○ (NOTE 3) |
| 6.21 | PMC axis control | Controls the PMC control axes. | ○ (NOTE 4) | ○ (NOTE 4) | × |
| 6.22 | Interlock along each axis | Applies interlocks along each axis and reads the travel axis and the direction of movement along the axis when the skip signal goes high. | ○ (NOTE 5) | ○ (NOTE 5) | × |
| 6.23 | Separation of P-CODE program UI/UO | Separates the system variables for macro DI/DO into the signals for user programs and those for P-CODE programs. | ○ | ○ | ○ |
| 6.24 | Referencing the common variables for custom macros | Reads the common variables for the user program from and writes them to the P-CODE program. | ○ | ○ | ○ |
| 6.25 | Displaying the conversational macro screen when power is turned on | Displays the conversational macro screen in the emergency stop state of the CNC or in the external reset state. | ○ | × | × |
| 6.26 | Masking of screen switching with the CUSTOM key | Inhibits the conversational macro program from being executed again from the beginning when the CUSTOM key is pressed on the conversational macro screen. | ○ | × | × |
| 6.27 | Searching for a P-CODE workpiece number | Directly executes the P-CODE program registered in an executable macro variable. | × | × | ○ |
| 6.28 | Function for calling a user program with an executable macro | Calls a user program from the P-CODE programs for executable macros. | × | × | ○ |
| 6.29 | Arithmetic function | Logarithmic function, Exponential function Arc sine function and Arc cosine function can be used. | ○ | ○ | ○ (NOTE 6) |
| 6.30 | MDI key image read function by conversational macro | MDI key image that reflects MDI key status (pressed/or released) can be read from the conversational macro. | ○ | × | × |
| 6.31 | Window function | Various CNC system information can be referred to through window. | ○ | ○ | × |
| 6.32 | Special character registeration and display function | Maximum 40 special characters can be registered and displayed. | ○ (NOTE 7) | × | × |
| 6.33 | Execution macro call mask function | An execution macro call can be masked by using an executor parameter or macro variable. | ○ | ○ | ○ |

○ : Usable Δ : Partially usable × : Unusable

**NOTE**

1 Since the Series 21 do not support the graphic display feature, they cannot use the drawing, painting, or other functions based on the graphic coordinate system.
2 This function can be used only for Series 16-B.
3 This function can not be used for Series 20 and for Series 21.
4 This function can not be used for Series 20-TA.
5 This function can not be used for Series 20.
6 To use the arithmetic function in the execution macro, custom macro B option must be combined in the CNC.
7 This function can be used only for Series 20.

**Table 6 (b) Functions of the Macro Executor and the Associated G Codes and Macro Variables (1/2)**

| No. | Function | Associated G code | Associated macro variable | Conversational | Auxiliary | Executable |
|---|---|---|---|---|---|---|
| 6.1 | Screen display function | G01, G02, G03, G202, G204, G206, G240, G242, G243, G244, G249 | #8509 | ○ (NOTE 1) | Δ (NOTE 1) | × |
| 6.2 | Address function | ——— | ——— | ○ | ○ | ○ |
| 6.3 | Writing at PMC addresses | G310 | | ○ | ○ | ○ (NOTE 2) |
| 6.4 | Reader/punch interface control | G330, G331, G335, G336, G337, G338, G339 | #8539 | ○ | ○ | × |
| 6.5 | Reading and writing of an NC program | G320, G321, G325, G326, G327, G328, G329 | #8520, #8521, #8522, #8523, #8529 | ○ | ○ | × |
| 6.6 | Continuous input with the cursor and page keys | ——— | #8501 | ○ | × | × |
| 6.7 | Masking the display of O and N numbers | ——— | ——— | ○ | × | × |
| 6.8 | Reading and presetting the cutting time and cutting distance | ——— | #8553, #8554 | ○ | ○ | × |
| 6.9 | Reading and presetting the relative coordinates | ——— | #8998, #8999 | ○ | ○ | × |
| 6.10 | Key/data input control | ——— | #8501, #8502, #8503, #8504, #8552 | ○ | × | × |
| 6.11 | Cursor control | ——— | #8505, #8506, #8507 | ○ | ○ | × |
| 6.12 | Processing of P-CODE variables of array type | G315 | #8511, #8512, #8513, #8514, #8515, #8516, #8517, #8519 | ○ | ○ | × |
| 6.13 | Torque limit control | ——— | #8621, #8622, #8623, #8624 | ○ | ○ | ○ |
| 6.14 | Reading of A/D converter data | ——— | #8631, #8632, #8633, #8634 | ○ (NOTE 3) | ○ (NOTE 3) | × |
| 6.15 | Key-in line function | ——— | #8561, #8562, #8563, #8564, #8565 | ○ | × | × |
| 6.16 | Reading the status indicating editing in the background | ——— | #8526 | ○ | ○ | × |
| 6.17 | Reading the number of registered programs | ——— | #8527 | ○ | ○ | × |
| 6.18 | Reading the unused capacity of CNC program memory | ——— | #8528 | ○ | ○ | × |
| 6.19 | Reading the remaining travel distance | ——— | #5181 to #5188 | ○ | ○ | × |
| 6.20 | Use of offset memory and extended system variables in the workpiece coordinate system | ——— | #100000 and up | ○ (NOTE 3) | ○ (NOTE 3) | ○ |
| 6.21 | PMC axis control | G340, G341, G344, G345, G346, G348, G349 | #8602, #8700<br>#8710 to #8713, #8715,<br>#8720 to #8723, #8725<br>#8730 to #8733, #8735<br>#8740 to #8743, #8745 | ○ (NOTE 4) | ○ (NOTE 4) | × |

○ : Usable Δ : Partially usable × : Unusable

**Table 6 (b) Functions of the Macro Executor and the Associated G Codes and Macro Variables (2/2)**

| No. | Function | Associated G code | Associated macro variable | Conversational | Auxiliary | Executable |
|---|---|---|---|---|---|---|
| 6.22 | Interlock along each axis | ——— | #8600, #8601 | ○ (NOTE 5) | ○ (NOTE 5) | × |
| 6.23 | Separation of P-CODE program UI/UO | ——— | ——— | ○ | ○ | ○ |
| 6.24 | Referencing the common variables for custom macros | ——— | #99000 | ○ | ○ | ○ |
| 6.25 | Displaying the conversational macro screen when power is turned on | ——— | ——— | ○ | × | × |
| 6.26 | Masking screen switching with the CUSTOM key | ——— | ——— | ○ | × | × |
| 6.27 | Searching for a P-CODE workpiece number | ——— | ——— | × | × | ○ |
| 6.28 | Function for calling a user program with an executable macro | ——— | ——— | × | × | ○ |
| 6.29 | Arithmetic function | ——— | ——— | ○ | ○ | ○ (NOTE 6) |
| 6.30 | MDI key image read function by conversational macro | ——— | #8549 | ○ | × | × |
| 6.31 | Window function | ——— | #8998, #8999 | ○ | ○ | × |
| 6.32 | Special character registeration and display function | ——— | ——— | ○ (NOTE 7) | × | × |
| 6.33 | Execution macro call mask function | ——— | #8690, #8691 | ○ | ○ | ○ |

○ : Usable  Δ : Partially usable  × : Unusable

**NOTE**

1 Since the Series 21 do not support the graphic display feature, they cannot use the drawing, painting, or other functions based on the graphic coordinate system.
2 This function can be used only for Series 16-B.
3 This function can not be used for Series 20 and for Series 21.
4 This function can not be used for Series 20-TA.
5 This function can not be used for Series 20.
6 To use the arithmetic function in the execution macro, custom macro B option must be combined in the CNC.
7 This function can be used only for Series 20.

## 6.1 SCREEN DISPLAY FUNCTION

In this section, the terms twelve–soft–key type and seven–soft–key type represent the following display units:

Twelve–soft–key type: Those display units with twelve (10 + 2) soft keys (such as the 14” CRT, 10” LCD, 9.5” LCD, and 10.4” LCD units)

Seven–soft–key type: Those display units with seven (5 + 2) soft keys (such as the 9” CRT, 8.4” LCD, 9” PDP, and 7.2” LCD units)

### 6.1.1 Coordinates System of Screen

The coordinates system used for character display and cursor display in the conversational macro program is called the character coordinates system.

The coordinate system used for graphic display in the conversational macro program is called the graphic coordinate system.

(1) Display unit with seven soft keys

The graphic coordinate system can be selected from the standard mode and the high resolution mode. Select one of these by the parameter HRGR (No.9003#2) in compiling. Select high-resolution mode for Series 16 and Series 18.

As for the character coordinates system, the horizontal direction is X coordinate and the vertical direction is Y coordinate. Specify the X coordinates from 0 to 39 from left to right and the Y coordinates from 0 to 15 from top to bottom. Any command of display exceeding the above range will be ignored. 1 unit is 1 character.

Since the 13th is used for the input section and the 14th is used for the state display section, they cannot be specified. Furthermore, the soft key frame section cannot be specified either.

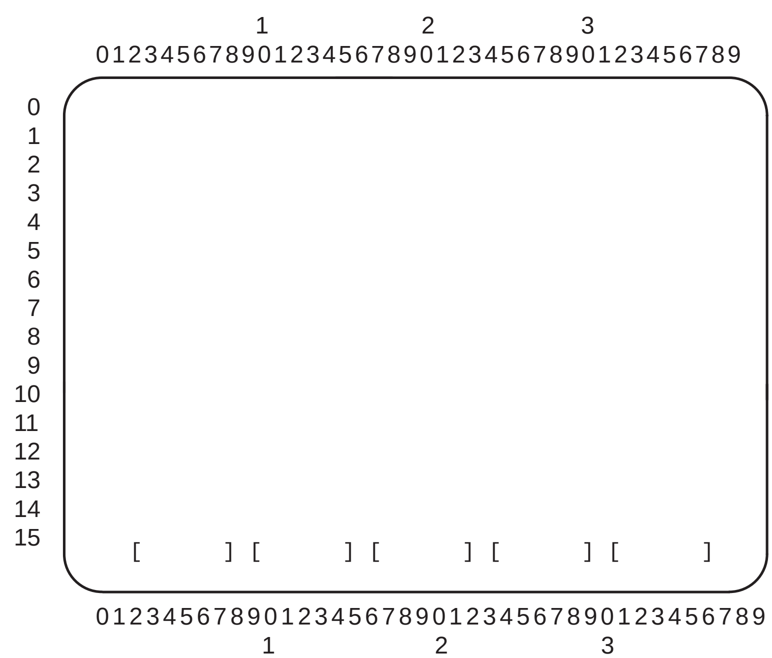



**Fig.6.1.1(a) Character Coordinate System (In Case of Seven Soft Keys)**

The coordinates system used for graphic display in the conversational macro function is called the graphic coordinates system.

As for the graphic coordinates system, too, the horizontal direction is the X coordinate and the vertical direction is the Y coordinate. Specify the X coordinates from -160 to 159(-320 to 319) from left to right and the Y coordinates from -112 to 112(-200 to 199) from bottom to top.(Value in parenthesis is for high resolution mode.) Any command of display outreaching the above range will be ignored. 1 unit is 1 dot.

The center of the screen will be (X, Y) = (0, 0).

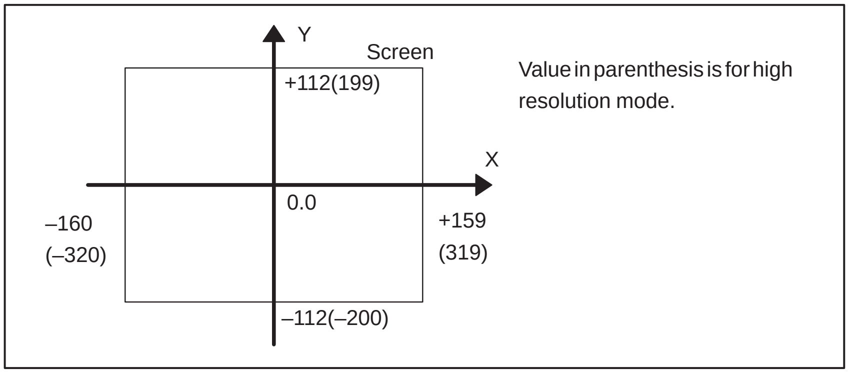



**Fig.6.1.1(b) Graphic Coordinate System**

(a) Standard mode

Use the same coordinate system as that of the series 0 standard seven soft keys.

Center of screen

(X coordinates, Y coordinates) = (0, 0)

Left and right of screen

(X coordinates) = (–160 – 159)

Top and bottom of screen

(Y coordinates) = (–112 – 112)

(b) High resolution mode

Use the high resolution coordinate system.

Center of screen

(X coordinates, Y coordinates) = (0, 0)

Left and right of screen

(X coordinates) = (–320 – 319)

Top and bottom of screen

(Y coordinates) = (–200 – 199)

Graphic display macro prepared in the series 0 standard seven soft keys can almost be used unchanged in standard mode. The high resolution graphic mode allows highly accurate display that was not possible in standard mode.

(2) Display unit with twelve soft keys

(a) Character coordinates line, graphic coordinates line

Character coodinate system instructs in the range of 0-79 for the left and righ directions (X coodinates), and 0-24 for the up and down directions (Y coordinates). The 20th line is used for input and the 21st line is used by the system for status display and can not be satisfied. The space for the software key also cannot be satisfied.

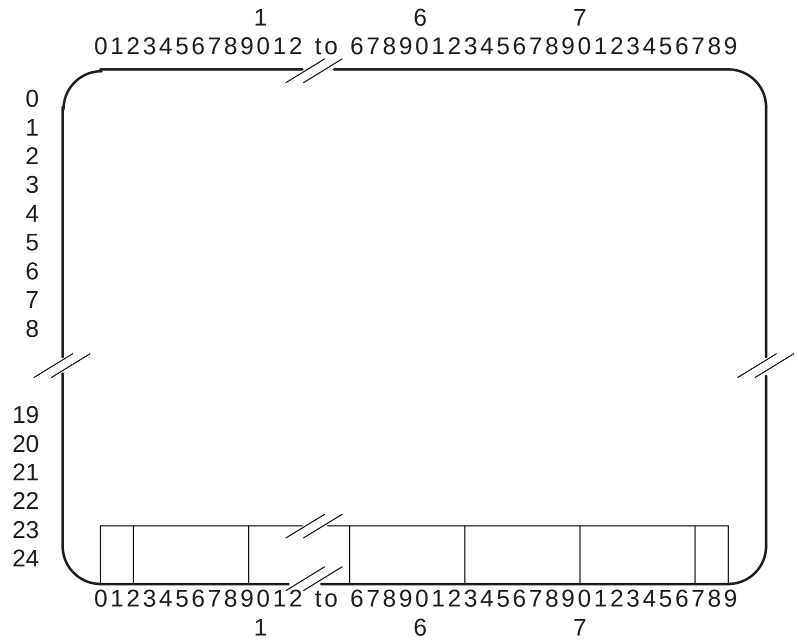



**Fig.6.1.1(c) Character Coordinate System (twelve soft keys)**

Instruct the following graphic coordinates of the graphic coordinates system.

Center of screen

(X coordinates, Y coordinates) = (0, 0)

Left and right of screen

(X coordinates) = (–320 – 319)

Up and down of screen

(Y coordinates) = (–200 – 199)

## 6.1.2 Screen Display Control Code

It is possible to make the screen freely by commanding the contents of the screen display with the conversational macro program. The following control codes are prepared for the screen display.

1. G243 Character display
2. G202 Screen erase
3. G240 Color specification
4. G244 Graphic line specification
5. G242 Setting of graph start point
6. G01 Straight line graph
7. G02 Circular graph (Clockwise)
8. G03 Circular graph (Counterclockwise)

G202, G242 and G244 are the one-shot G codes. All the other G codes are the modal G codes and they are considered as belonging to the same G code group.

Also the following are the modal addresses.

X : X coordinate of the character coordinates system, X coordinate of the graphic coordinates system

Y : Y coordinate of the character coordinates system, Y coordinate of the graphic coordinates system

I : X coordinate at the center of the circular graph of the graphic coordinates system

J : Y coordinate at the center of the circular graphic of the graphic coordinates system

A : Character size (Character display)

B : Flash specification (Character display)

F : Format of numeric display (Character display)

Z : Zero supress specification of numeric display (Character display)

Q : Circular graph (Graphic display), PC address write data

**NOTE**

Since the Series 21 do not support the graphic display feature, they cannot use G01, G02, G03, G242, or G244. Only character deletion is possible when using G202 (screen deletion).

(1) Character display (G243)

G243 Xx Yy Aa Bb Cc (c..) (*hhhh..*) Kk Ff.e Zz Dd Pp ;

There are 2 ways of commanding the character string to display.

a) Command to display a character string by enclosing it with parentheses:
G243 (FANUC 16-SERIES) ; ”FANUC 16-SERIES” will be displayed.

Those which you can command by enclosing with parentheses are the alphabet, numbers, minus marks, and decimal points only.

b) Commandto display a character string by bounding with ”(*” and ”*)” by internal codes:

G243 (*46 41 4E 55 43*) ; ”FANUC” will be displayed.

G243 (*4E 43 4175 4356*) ; ”NC system” will be displayed.

Command with hexadecimal number. Divide the internal codes with blanks. Take a space for 2 characters aside for hiragana and kanji. The coordinates will be renewed when a character string is displayed.

For the character string, it is possible to specify up to maximum total 255 characters in 1 block. Count hiragana and kanji as 2 characters respectively.

Also, it is possible to command the character code to display directly at the address C by decimal number. The commandable codes are 32 - 95 (20 - 5F with hexadecimal number) or 160 - 223 (A0 - DF with hexadecimal number).

C40 → * C61 → =

Command the display positions of the character strings at the addresses X and Y with the character coordinates.

Command character size at the address A.

A1 = Normal size

A2 = Full size

A3 = Triple size (3 × 2 size)

A4 = 2 × 2 size

The following characters and symbols can be displayed at full size:

Alphabetic characters, numeric characters, kana characters, blanks, the plus sign (+), minus sign (–), period ( . ), equal sign (=), comma ( , ), asterisk ( * ), slash ( / ), paretheses ( ( ) ), square brackets ( [ ] ), inequality signs ( < and > ), and the sharp (#)

The triple size is 3 times as wide as and 2 times as long as the normal size. The characters which can be displayed with the triple size characters are the alphabet, numbers, minus marks, decimal points and blanks. No other characters can be displayed with the triple size.

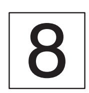
A 2-by-2 character is two times taller and two times wider than a standard character. Fifteen types of 2-by-2 characters can be displayed: numbers (0 to 9), plus sign, minus sign, decimal point, asterisk, and slash. Note, however, that the Greek character ϕ is displayed for a slash.

**NOTE**

Only the Series 20 allows 2-by-2 character display. With the Series 16/18/21, 2-by-2 characters cannot be displayed; A4 cannot be specified.

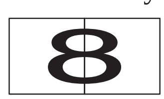
1. Standard character

   G243 Xx Yy A1 (8)



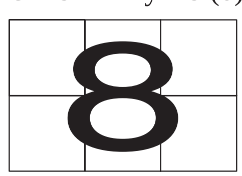

2. Full-size character

   G243 Xx Yy A2 (8)



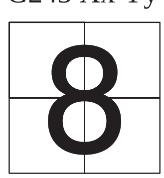

3. Triple-size character (3-by-2 character)

   G243 Xx Yy A3 (8)




4. 2-by-2 character

   G243 Xx Yy A4 (8)



Bink control is commanded by the address B.

B0 = Does not blonk.

B1 = Blinks slowly. (Energized for about 1/2 sec. and deenergized for about 1/4 sec..)

B2 = Blinks quickly. (Energized for about 1/4 sec. and deenergized for about 1/8 sec..)

> **NOTE**
> When a blink is specified, the display is energized or deenergized according to the timer condition when the display is commanded. Therefore, when the display is not repeated, it will be kept energized or deenergized.

Command the number of blanks at the address K. The blanks as many as commanded at K will be displayed. When the blanks are displayed, the coordinates will be renewed.

Command the format to display the numeric values at the address F. Command the number of display figures on the left of the point and the number of figures under the point on the right of the point.

Command at the address Z whether you do ”leading zero suppress” or not when displaying the numeric values.

Z = 0 .. leading zero suppress will take place.

= 1 .. leading zero suppress will not take place.

The mark will not be displayed when Z is 0.

Command the numeric values to display at the address D.

Command the sequence No. that the character string is cataloged at the address P. The character displayed will be the character string in one block of the sequence No. commanded with P in the program set at the character string registered program control variable #8509. The variable for controlling character-string registration programs (#8509) is defined as the number of the first program in the character-string registration program group. Use the five digits for address P to select the desired character-string registration program in the group.

G243 Xxx Yyy Ponnnn;

o : A number from 0 to 8 to select a program in the character-string registration program group

nnnn : Sequence number 0001 to 9999

Example 1) #8509 = 1000 ;

G243 P10 ; ⇒ Displays the character string with sequence number N10 in program O1000

G243 P80010 ; ⇒ Displays the character string with sequence number N10 in program O1008.

Up to nine character-string registration programs can be selected as desired in the above way.

Example 2)

| O9000; | O8000; |
|---|---|
| : | : |
| #8509=8000; | N10(IJK); |
| X0Y0; | |
| G243(ABC)P20; | N20(XYZ); |

In this case, one block of the sequence No. 20 in the program No. 8000 will be executed.

In the above example, ”ABCXYZ” will be executed.

And address P executes the specified block after the process of the corresponding block is finished. Consequently, ”ABCXYZ” will be displayed though command is done as follows with the above example.

G243 P20 (ABC) ;

Nothing but the character string will be described at the block specified by P.

When #8509 is 0, the block of the sequence No. specified in the current program will be executed.

**NOTE**

Process sequence of modal address

Program is processed by block in the normal NC program, but it is processed in sequence of address command in the conversation macro program.

(Example)

1 F8.3 ;
  G243 F5.1 D#100 ;
2 F8.3 ;
  G243 D#100 F5.1 ;

In 1 #100 is displayed with F5.1, but in 2 it is displayed with F8.3.

(2) Screen erase (G202)

G202 XxYyIiJjPp;

X = Start point of X coordinate in character coordinate

Y = Start point of Y coordinate in character coordinate

I = Number of characters to be erased from the start point (X coordinate) for partial erasing.

J = Number of characters to be erased from the start point (Y coordinate) for partial erasing.

P = 1 ... Erasing graphic screen

= 2 ... Erasing character screen

= 3 ... Erasing graphic and character screens

Also to make partial erasing, all X, Y, I and J addresses must be specified. If any address is omitted, all the screen is erased.

(3) Displayed color specification (G240)

G240 Pp $L_l$ ;

Colors of segments and character strings specified by conversational program can be designated.

P = 0 .. Black
= 1 .. Red =–1 Red (highlights)
= 2 .. Green =–2 Green (highlights)
= 3 .. Yellow =–3 Yellow (highlights)
= 4 .. Blue =–4 Blue (highlights)
= 5 .. Purple =–5 Purple (highlights)
= 6 .. Blueish green =–6 Bluish green(highlights)
= 7 .. White =–7 White (highlights)

L = 0 .. not blink
= 1 .. blink

If no designation is made, a segment and a character string are white without blink.

(4) Graphic line specification (G244)

G244 Pp ;

P = 0 .. Full line
= 1 .. Broken line
= 2 .. 1-dot line
= 3 .. 2-dot line
= 4 .. Erase

The graphic line is full line at power ON.

(5) Setting of graph start point (G242)

G242 Xx Yy ;

Command the display start point with the absolute coordinates at the addresses X and Y.

Next display will start at this point.

(6) Straight line graph (G01)

G01 Xx Yy ;

Command the coordinate of the end of the straight line with the absolute coordinate at the addresses X and Y.

A straight line will be made with the line commanded with G244 from the current position to the commanded coordinate. The coordinates will be renewed.

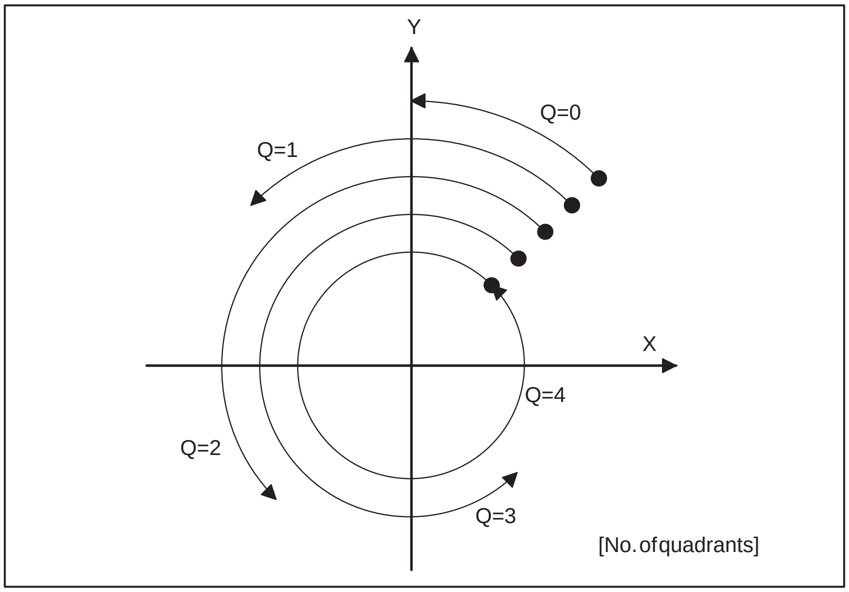
(7) Circular display (G02, G03)

G02 Xx Yy Ii Jj Qq ;

G03 Xx Yy Ii Jj Qq ;

G02 display the arc clockwise and G03 counter-clockwise.

Command the end of the arc with the absolute coordinate at the adresses X and Y.




Command the center coordinate of the arc with the absolute coordinate at the address I and J.

Command the number of quadrant (0 - 4) of the arc at the address Q. The arc will be graphed with the line commanded with G244 from the current position to the coordinate of the commanded end.

The coordinate will be renewed.

## 6.1.3 Graphic Painting Function

The graphic function for conversational macros enables painting the specified field.

> **NOTE**
> The graphic painting function cannot be used for Series 21.

(1) Description

The field to be painted is drawn using a solid line with the graphic function. Then the paint command is used together with control code G206 to specify any point inside the field and the boundary color. The field is filled with the color which is used as standard when G206 is issued.

G206PpXxYy;

Painting boundary color (P)

p = 0: Same as the fill color

p = 1: Red

p = 2: Green

p = 3: Yellow

p = 4: Blue

p = 5: Purple

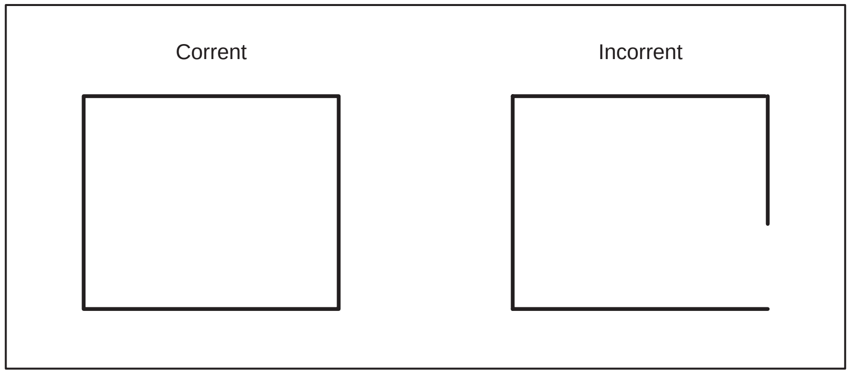
p = 6: Blue-green

p = 7: White

p = 8: Specified two or more colors other than black

Arbitrary point inside the field to be painted (X, Y)

> **NOTE**
> The field to be painted must have a closed boundary.



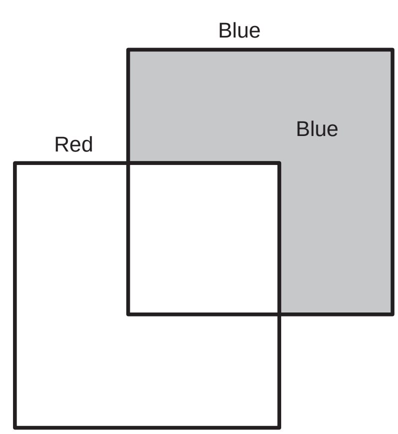

> **NOTE**
> To use the painting command on the high-resolution 9” monochrome CRT, set HRGC of compilation parameter 9004 to 1 to make the CRT enter the intensity modulation mode.

(2) Examples

(a) When p = 8 is specified

The innermost field is painted.

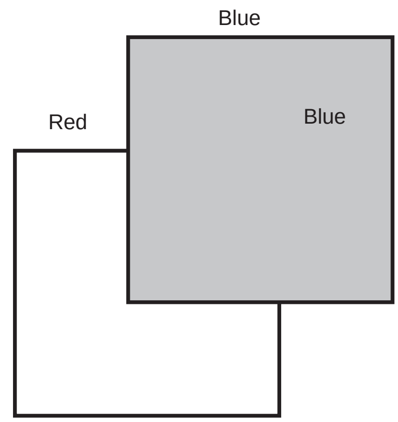
```
G240P1;
G244P0;
G242X0Y0;
G01X100;
Y100;
X0;
Y0;
G240P4;
G242X50Y50;
G01X150;
Y150;
X50;
Y50;
G206P8X110Y110;
```




(b) When the same color is specified for the fill and the boundary (p = 0)

A line with another color in the field to be painted is painted.

```
G240P1;
G244P0;
G242X0Y0;
G01X100;
Y100;
X0;
Y0;
G240P4;
G242X50Y50;
G01X150;
Y150;
X50;
Y50;
G206P0X110Y110;
```




## 6.1.4 Graphic Cursor Function

Graphic cursor can be displayed in the conversational macro.

(Command Format)

<u>G249 Pp Xxxx Yyyy;</u>

p : Control code
=0 : Display on (lit)
=1 : Display on (Blinks at low speed)
=2 : Display on (Blinks at high speed)
=3 : Display off

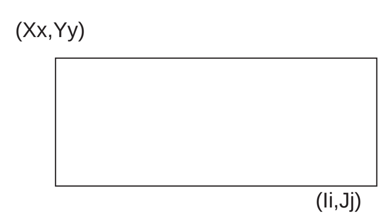
xxx/yyy : Graphic cursor display position (Graphic coordinate) (When display off is specified, this specification is ignored. It can be omitted).

## 6.1.5 Rectangular Display Function

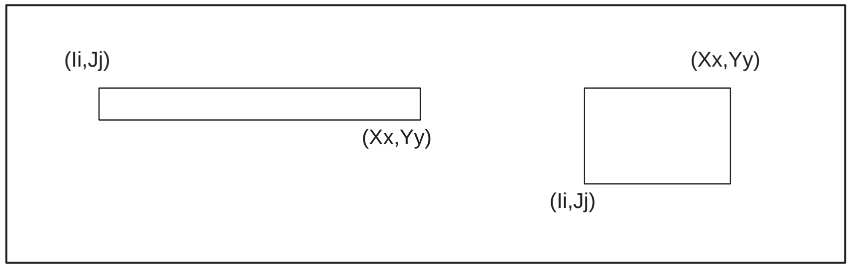
In a conversational macro, a rectangular figure can be displayed.

(1) Function

In rectangular display, a rectangle in which addresses X and Y, I and J become diagonal positions is displayed.




Addresses X and Y, I and J are any position that can create a rectangle.




(Command format)

G204 X_ Y_ I_ J_ [P ];

X : X coordinate of the start of a rectangle

Y : Y coordinate of the start of a rectangle

I : X coordinate of the end of a rectangle

J : Y coordinate of the end of a rectangle

P : Painting control = 1 : Painting in rectangle (red)
= 2 : Painting in rectangle (green)
= 3 : Painting in rectangle (yellow)
= 4 : Painting in rectangle (blue)
= 5 : Painting in rectangle (purple)
= 6 : Painting in rectangle (bluish green)
= 7 : Painting in rectangle (white)

Command in parentheses can be omitted.

The color of boundary of a rectangle is of specified by G240P_ and painted color in rectangle is of specified by address P.

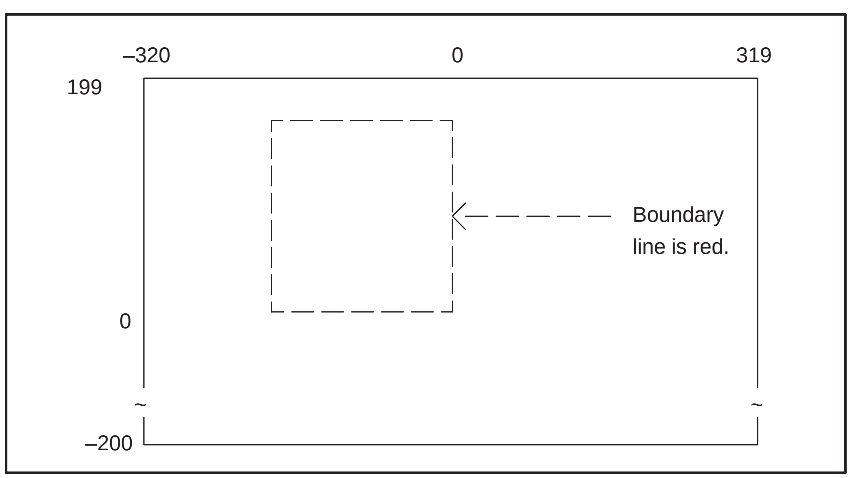
When address P is omitted, only boundary of a rectangle is displayed.

After the painted color is specified by P, the color remains the following movement. That is, the color specified by P changes the P value of G240P_.

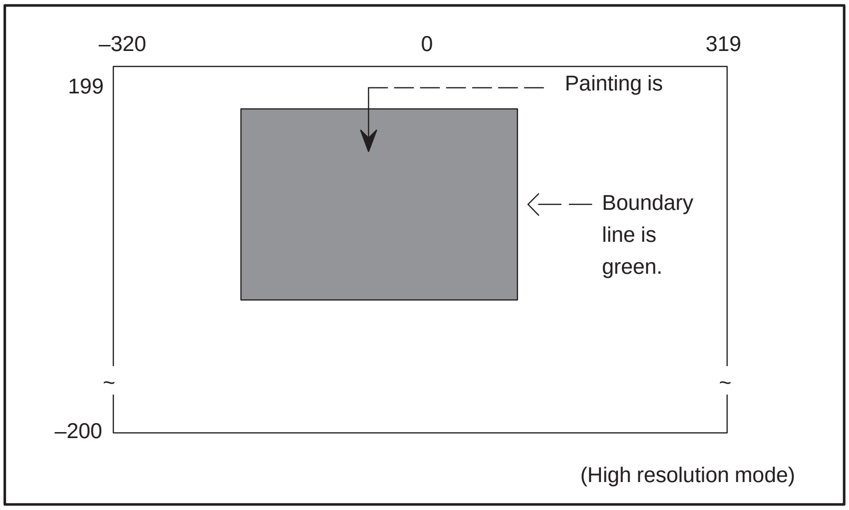
Type of line of rectangle is of a type specified by the line type specification command (G244P_).

(2) Example

G244 P1;
G240 P1;
G204 X-200 Y150 I0 J10;



G244 P0;
G240 P2;
G204 X50 Y50 I-200 J150 P1;




**NOTE**
Be sure to specify full line when you specify painting.
Painting is started from the point determined to be the midpoint between addresses X and I and that between addresses Y and J.
In painting, a boundary line can be displayed in any color other than black. Painting is executed only when specified on a black ground.

### 6.1.6 Intensity Modulation Mode Display of 9” Monochrome CRT

For 9” monochrome CRT, the conversational macro screen can be displayed in intensity modulation mode using a compile parameter (No. 9004#4, 9008#7).

Color specification code (G240) specifies intensity. During the monochrome mode, the color specification code is ignored.

G240 Pp L1;

P = 7 (Brightest ) to 1 (Darkest), 0 (Black)

L = Reverse specification

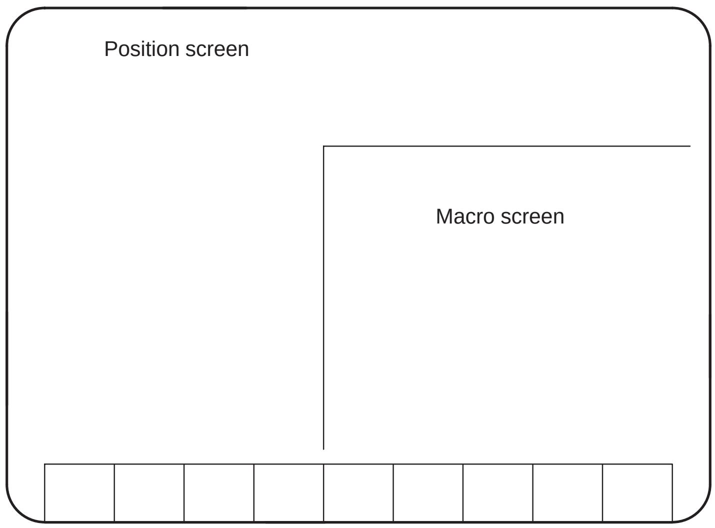
### 6.1.7 Displaying Seven Soft Keys Data on the Twelve Soft Keys Type

Data for a twelve soft keys can be displayed on the user-1 screen for a conversational macro screen on the twelve soft keys by setting US19W of compilation parameter (bit 5 of parameter 9006). (This screen is equivalent to the AUX screen in Series 0.)

The whole of user screens 2 and 3 can be used as macro screens while they are displayed. (The absolute coordinates are always displayed on the position screen.)

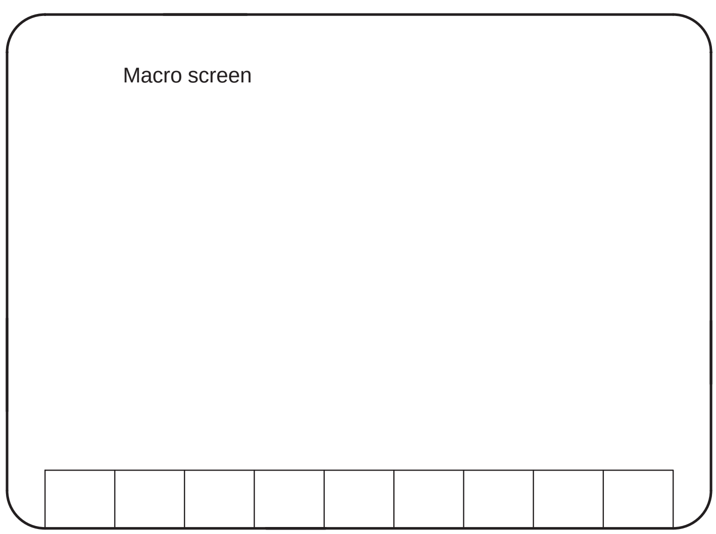
US19W = 1: Displays seven soft keys data on the twelve soft keys type

<<USER1 screen when seven soft keys data is displayed>>




**NOTE**
Upper left corner of the macro screen is X=40 and Y=7.

<<USER2 and USER3 screens>>



## 6.1.8 Function Screen Control Function

Writing a desired value in parameter 8510 enables displaying any function screen. Reading the value of parameter 8510 enables checking which function screen is currently displayed.

The following table shows the correspondence between function screens and the values of parameter 8510.

| Function screen for a small keyboard | Value of parameter 8510 | Function screen for a standard/FAPT keyboard | Value of parameter 8510 |
|---|---|---|---|
| POS : Current value display screen | 0 | POS : Current value display screen | 0 |
| PROG : Program display screen | 1 | PROG : Program display screen | 1 |
| OFFSET SETTING : Offset/setting screen | 2 | OFFSET SETTING : Offset/setting screen | 2 |
| SYSTEM : Parameter/diagnosis and suchlike | 3 | SYSTEM : Parameter/diagnosis and suchlike | 3 |
| MESSAGE : Alarm/message screen | 4 | MESSAGE : Alarm/message screen | 4 |
| CUSTOM GRAPH : Graphic/user screen | 5 | GRAPH : Graphic screen | 5 |
| CUSTOM : User screen | 6 | CUSTOM : User screen | 6 |
| | | FAPT : FAPT screen (FAPT key) | 7 |

## 6.1.9 Function for Masking the Status Display on the Conventional Macro Screen

The mode and status display can be masked on the user-1, -2, and -3 screens for displaying the conversational macro screen by specifying compilation parameter STDM (bit 2 of parameter 9006).

The 14th line on the seven soft keys and 21st line on the twelve soft keys can therefore be controlled with conventional macros.

## 6.1.10 Shift Function for Adjusting the Graphic Screen

The graphic coordinate system can be shifted in units of dots on the conversational macro screen by specifying compilation parameters 9048 and 9049.

Compilation parameter 9048:
Shift value along the X-axis in the graphic coordinate system on the conversational macro screen

Compilation parameter 9049:
Shift value along the Y-axis in the graphic coordinate system on the conversational macro screen

## 6.2 ADDRESS FUNCTIONS

The address functions are the functions to return the address of PMC or the contents of the parameter as the function values. However, as the address functions cannot be written, it is impossible to describe them on the left side of an operation. Control code G310 is prepared for PMC address writing.

The format of the address functions is as follows.

<Address><Address No.> or <Address><Address No.>.<Bit position> (parameter cannot be used) or<Address><Address No.>.<Axis No.>

Address P indicates the parameter and the other addresses indicate PMC addresses.

P100 returns the contents of the parameter No. 100 as function values respectively.

Since the bit position cannot be specified for parameters, if it is required, take the bit data using AND instruction.

Also, G100.1 returns the contents of the bit 1 of PMC address G100 ,and G105 returns the contents of PMC address G105 as function values respectively.

It is possible to describe <Address No.> and <Bit position>or<Axis No.> with variables or
# [<Expression>] or [<Expression>], instead of numbering them with numerical values directly.

For example, it is possible to describe as follows.

P#100

P#100.#101

G# [#100+1]

G [#100+1] .[[#100-1]/2]

The usable addresses are ”P” for parameter and X, Y, F, G, R, D,T,C, K and A of PMC addresses.

Format conforms to that used for ladder diagram.

Refer to the operator’s manual for the detail of parameters and to the PMC programming manual for details of the PMC addresses.

**NOTE**
Refer to the section titled ”PMC Address Write Control Code (G310)” for writing data to the PMC addresses R,D,C, and K.

## 6.3 READING AND WRITING A PMC ADDRESS

(1) PMC address write control

Data of one, two, or four bytes can be written when address L is specified with control code G310.

G310 Dd Qq Lx;

G310 Rr Qq Lx;

G310 Cr Qq Lx;

G310 Kr Qq Lx;

These commands write the data indicated by address Q at the PMC address indicated by address D, R, C, or K, a number of bytes at a time as indicated by address L. The data cannot be specified in bits.

The write data indicated by address Q is rounded as follows: Decimal places of 0.5 and over are counted as a whole number and the rest are discarded. Then the data is expressed in binary form and written. When the data is a negative value, it is handled and converted as two's complement.

When the following is specified, the data is written in the PMC data area (D300 to D303) as shown in the table below:

#100 = –500.0;

G310 D300 Q#100 L4;

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| D300 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| D301 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| D302 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| D303 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

In two's complement, the decimal number –500 becomes FFFFFE0CH in hexadecimal.)

When the specified data contains more bytes than that indicated by address L, only the data of the length specified by address L is written. This will not cause an error. (If L1 is specified in the example above, the identical data is written only in D300.)

(2) PMC address read control

When a variable is indicated by address P with control code G310, the PMC data can be read. Address L indicates that the data of one, two, or four bytes is read at a time.

G310 Dd Pp Lx;

G310 Rr Pp Lx;

G310 Cr Pp Lx;

G310 Kr Pp Lx;

These commands read the data which starts from the PMC address indicated by address D, R, C, K or A and whose length is indicated in bytes by address L into the variable indicated by address P. The read data is handled as binary data consisting of the specified number of bytes, converted, and assigned to the variable.

If G310 D400 P101 L2; is specified when the PMC data area (D400 to D401) has the data shown below, –500.0 is input in #101.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| D400 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| D401 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

The address function can read data the specified number of bytes at a time. However, the following must be noted: As the address function handles and converts the read data as a signed value (in two's complement), the read data may be changed.

If the following is specified with the example shown above, –2.0 and 254.0 are read into #102 and #103 respectively.

G310 D401 P102 L1;

#103 = D401;

**NOTE**

1 When address Q is specified, write control is executed. When address P is specified, read control is executed. If both addresses Q and P are specified, write control is executed.
2 Addresses other than D, R, C, K (X, Y, G, F...) cannot be specified.
3 With address L, 1, 2, or 4 can be specified. If another value is specified, or if address L is omitted, the data is written or read the specified number of bytes at a time.
4 For details of the PMC address, refer to the PMC programming manual.

# 6.4 READER PUNCHER INTERFACE CONTROL BY CONVERSATIONAL MACRO

## 6.4.1 Outline

The conversation macro function allows the reader/puncher interface to be controlled.

The control is performed by combining the following four control codes.

When this function is used, always set the expansion function validity (No. 9002.7, EXT1) to 1 in the parameter on compiling.

G330 → reader puncher interface open
G331 → reader puncher interface close
G335 → 1 byte read
G336 → Data transmit
G337 → input of macro variable
G338 → output of macro variable

Select either channel 1 or channel 2 as a circuit to be used by specification on opening.

Channels 1 and 2 cannot be controlled simultaneously.

In addition, when the normal I/O function is commanded on the channel which is being used in this function, operation is not proper if the channel, where the input of NC program is being executed by the I/O function, is opened.

Select one control method out of the three methods shown in the following by specification on opening a circuit.

(1) User macro control

The reader puncher interface is opened in the bidirectional mode and no output control of control codes (DC1 to DC4) can be carried out by the macro executor.
Use the macro executor when creating your own protocol.
When overflow of the reception buffer is detected, send the transmission stop/restart message to the target device by switching ON/OFF the control signal ”RS”.

(2) Read control (DC1/DC3 automatic control)

The ”DC1” code is automatically sent on opening, thus requesting the target device to transmit data.
The ”DC3” code is transmitted on closing.
In addition, when overflow of the reception buffer is detected, the output of ”DC3” and ”DC1” is automatically controlled.
No control of data transmission (G336) can be performed when the reader/puncher interface circuit is opened in the read control mode. (End code = 8)

(3) Transmission control (DC2/DC4 automatic control)

The ”DC2” code is automatically on opening, thus requesting the target device to transmit data.
The ”DC4” code is transmitted on closing.
The interruption and restart processing of transmission by the target device using the ”DC3” and ”DC1” are also automatically carried out.

No control of 1 byte reading (G335) can be carried out when the reader/puncher interface circuit is opened in the transmission control mode (End code = 8). There is an end code to check to see if the commanded control code has been properly executed or not.
Check the end code after executing G330 to G338.

#8539 : End code (only for reading)

However, the circuit closing processing always ends normally.

## 6.4.2 Function Details

(1) Circuit open (G330)

G330 Pp Bb Ss Cc ;

The reader puncher interface of the specified channel is opened according to the control method and control conditions to enable it to be used.

Select the channel and control method used by the specification of address ”P”.

1: Channel 1 User macro control

2: Channel 2 User macro control

11 : Channel 1 Reading control (DC1/DC3 automatic control)

12 : Channel 2 Reading control (DC1/DC3 automatic control)

21 : Channel 1 Transmission control (DC2/DC4 automatic control)

22 : Channel 2 Transmission control (DC2/DC4 automatic control)

**NOTE**
In case of 21-TB, only channel 1 is used.

Specify the baud rate by the specification of address ”B”.

| | | | | | |
|---|---|---|---|---|---|
| 1 : | 50b/s | 2 : | 100b/s | 3 : | 110b/s |
| 4 : | 150b/s | 5 : | 200b/s | 6 : | 300b/s |
| 7 : | 600b/s | 8 : | 1200b/s | 9 : | 2400b/s |
| 10 : | 4800b/s | 11 : | 9600b/s | | |

Specify the stop bit and parity by the specification of address ”S”.

1 : 1 stop bit, No parity

2 : 2 stop bits, No parity

11 : 1 stop bit, Odd parity

12 : 2 stop bits, Odd parity

21 : 1 stop bit, Even parity

22 : 2 stop bits, Even parity

Specify the output code on transmitting data by the specification of address ”C”.

1 : Output code (ASCII)

2 : Output code (ISO)

Example) Device and transmission control of channel 2
(DC2/DC4 automatic control)

Baud rate = 4800 b/s, Stop bit = 2, Even parity, ISO code output

G330 P22 B10 S22 C2 ;

IF [#8539 NE 0] GOTO 900 ;

Open processing end

N900 error processing :

(2) Circuit closing (G331)

G331 ;

The circuit, which is currently open, is closed.

There is no end code for the circuit closing processing.

It always ends normally.

(3) 1 byte reading (G335)

G335 Pp ; p : Reading variable number

Data 1 byte received through circuit is read into the variable area specified.

The received data is stored in the reception buffer (128 bytes) temporarily and is read one byte at a time by this control code.

When there is no reception data, end code 255 is sent.

If the reception buffer is nearly overflowing due to delay of reading processing for the reception data, either of the following processing is carried out according to the opening specification of control method.

(a) In the case of user macro control

When overflow of the reception buffer is detected, the transmission stop is requested to the target by the turning OFF of control signal ”RS”.
Then, when there is an empty reception buffer area by reading, the signal (RS) is turned ON, thus requesting the target device to continue transmission.

(b) In the case of reading control (DC1/DC3 automatic control)

When overflow of the reception buffer is detected, the ”DC3” code is output automatically, thus requesting the target device to stop transmission.

Then, when there is an empty reception buffer area by reading, the DC1” code is output, thus requesting the target device to continue transmission.

(c) In the case of transmission control (DC2/DC4 automatic control)

When the circuit is opened in the transmission control mode, no control of 1 byte reading is carried out (end code = 8).

Example)

N100 G335 P100;

IF [#8539 EQ 255] GOTO100;
IF [#8539 NE 0] GOTO900;
Reading processing ended
(#100: Received data)

N900 error processing

(4) Data transmission (G336)

G336 transmission data;

The following commands of transmission data are available:

(c...) → String command 1

(∗hh..∗) → String command 2

Ff → Data format command

Zz → Zero suppress command

Dd → Data command

Pp → Block number command

Kk → Null code command

Cc → 1-character command

See screen display control code G243 (character display) for string commands 1 and 2 and address F, Z, D, P, and K commands since they are similar.

The commanded string is converted into either ASCII or ISO code and is transmitted by the specification on opening.

However, the code output by address K is not the space but the null code (no punching).

In the one-character command by address C, no code conversion processing is carried out.

Use it when the code is output other than the control code (DC1 - DC4) or ASCII/ISO code.

Example)

```
G336 C18 K20 (FANUC) K20 C20;
IF [#8539 NE 0] GOTO900;
Transmission processing ended
N900 error processing;
```

The following data is transmitted in the circuit by this command:

C18 → DC2 (12hex)

K20 → Null code 20 (Feed)

(FANUC) → ”FANUC” (ASCII/ISO)

K20 → Null code 20 (Feed)

C20 → DC4 (14hex)

When the circuit is opened in the reading control mode, no control of data transmission is carried out (End code = 8).

### 6.4.3 Inputting and Outputting Macro Variables

Macro variable data input and output can be conducted simply by an instruction via reader puncher interface. Instruct the control with the following G code.

G337 → macro variable data input

G338 → macro variable data output

An end code #8539 is provided to check that input and output processing runs correctly or not. Check the end code #8539 after running G337, G338.

#8539 → end code (read only)

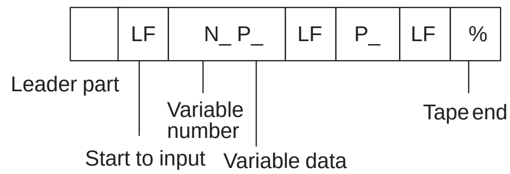
(1) Macro Variable Data Input (G337)

G337 Pp ;

p : the head variable number which mounts the macro variable (omissible)

A macro variable data, which is input from reader puncher interface that opened with the read control mode, is read and set it to the correspondenting variable.

The tape format of macro variable data is as follows;




Information before ”LF” which appears on the tape at first is all disregarded.

The tape to the end (”EOR”) from the first ”LF” is analyzed as significant information.

A section which is spaced with ”LF” and ”LF” into significant information segments called a block, and one block shows one macro variable data.

Address ”N” can be omitted. In this case, the variable number is interpreted to be (the variable number of the preceding block) +1. If ”N” is omitted at the head block, the variable number is assumed to be the one specified by address ”P” with instruction G337. Therefore, a tape without ”N” is made out, and arbitrary variables are read by the instruction of ”337” ” Pp”.

Address ”P” on the tape indicates the value of the variable and cannot be omitted. ”LF” is specified after ”P” for value zero (#0) without specifying the value.


”LSK” or ”EDIT” goes on and off under the right of the screen if the data is being read. If the data is being read, the screen cannot be changed.

**NOTE**

1 G337 is a one shot G code.

2 Address ”O” data and every ”CR”, space, ”DEL” code is disregarded within the significant information section.

3 Only ISO or ASCII code can be used for input code.

4 TV, TH check does not run.

(2) Outputting Macro Variable Values(G338)

G338 Pp Qq Ff Zz ;

A specified macro variable is changed into a prescribed tape format and output from reader puncher interface which opened with the transmit control mode.

Output code is selected between ISO and ASCII code by the open command.

- p : Specify the head of output macro variable
- q : Specidy the number of output macro variable data
- f : Specify the output format of macro variable data (modal value when omitted)
- z : Specify the zero-suppress of macro variable data (modal value when omitted)

For output tape format, same as the input format, the head variable number address ”N” and variable values with address ”P” at the head block, then variable values are continuously output with P in the following blocks and finally, the tape end (”EOR”) code is output.

”CR” code of every block can be output by the compile parameter PTCR(No.9003#6) . Use this code to make carriage return of a printer, etc.

PTCR=0: ”CR” is not output after ”LF”, when outputting macro variable values.

| | | | | | | |
|---|---|---|---|---|---|---|
| % | LF | N10000P1234 | LF | P5678 | LF | % |

PTCR=1: ”CR” is output twice after ”LF”, when outputting macro variable values.

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| % | LF | CR | CR | N10000P1234 | LF | CR | CR | P1234 | LF | CR | CR | % |

Control of the feed part does not act at output. Conduct with ”G336Kk” data transmission to feed with a paper tape puncher, etc.

”EDIT” goes on and off at the lower right of the screen if a data is outputting. Screen cannot change if data is outputting.

**NOTE**

1 G338 is a one-shot G code.
2 Only ISO code or ASCII code can be used for output.
3 An end code is communicated by 115 when an invalid data is specified to a variable number, and the outputting stops.

### 6.4.4 Extending the Function for Inputting and Outputting a Macro Variable

The conversational macro can input and output a macro variable via the reader/punch interface. This function can be extended so that two or more variable groups can be output at a time and the output data can be read into another variable.

(1) Outputting macro variables

By adding address R to G338 described in (2) of Section 6.4.3, the tape format of the variable data to be output can be controlled.

(Command format)

G338 Pp Qq Ff Zz Rr;

-------------

R0 : Standard format (Same as when R is not specified)

R1 : Variable numbers are not output.

R10 : At the end of the tape, % (EOR) is not output.

R11 : Variable numbers are not output. At the end of the tape, % (EOR) is not output.

R20 : At the beginning of the tape, % (EOR) is not output.

R21 : Variable numbers are not output. At the beginning of the tape, % (EOR) is not output.

R30 : At the beginning and end of the tape, % (EOR) is not output.

R31 : Variable numbers are not output. At the beginning and end of the tape, % (EOR) is not output.

(a) When variable numbers are not output, the function for inputting a macro variable (G337) can read the output variable data into a variable indicated by address P.

(b) When % (EOR) is not output at the beginning and/or end of a tape, two or more variable groups can be output to form a single tape format.

(Example)

G330 Pp Bb ... ;

| | |
|---|---|
| G338 P10000 Q10 F8.3 Z1 R11;⇒ IF[#8539NE0]GOTO999; | At the beginning, % (EOR) is output. Then the data of #10000 to #10009 is output without variable numbers. At the end, % (EOR) is not output. |
| G338 P11000 Q20 F8.3 Z1 R31;⇒ IF[#8539NE0]GOTO999; | At the beginning and end, % (EOR) is not output. The data follows the data output above to form a single tape format. |
| G338 P12000 Q10 F8.3 Z1 R21;⇒ IF[#8539NE0]GOTO999; | At the end, % (EOR) is output. The tape format is closed. |

G331;

```
%;P___;P___; .. ;P___;P___;P___; ... ;P___;P___;P___; ... ;P___;%
|←#10000 to #10009→|←#11000 to #11019→|←#12000 to #12009→|
      ”R11”                ”R31”              ”R21”
```

(2) Function for inputting macro variables

By adding address Q to G337 described in (1) of Section 6.4.3, the number of variables to be input can be specified. When address R is also specified, data reading can be continued.

(Command format)

G337 Pp Qq R99;

---------

P : Number of the variable to be read (Valid when variable number N is not specified on the input tape)

Q : Number of variables to be read (Can be omitted)

R99 : Reading to be continued (Can be omitted)

(a) When address Q is specified, the number of variables to be read can be specified. After the specified number of variables are input, completion code #8539 is set to 99 to indicate that data reading can be continued. If % (EOR) is read before reading the specified number of variables, completion code #8539 is set to 0. If address Q is omitted, the system assumes that an infinite number is specified.

(b) When R99 is specified, the remaining variable data can be read after the specified number of variables are read and completion code #8539 is set to 99. If data reading is continued without R99, the data of a variable is lost in the middle of reading because of a significant information check. (The data up to the first ”;(LF)” is skipped.)

(Example) Reading the tape output according to the example described above

G330 Pp Bb ... ;

| | |
|---|---|
| G337 P100 Q10 ;⇒<br>IF[#8539 NE 99]GOTO888; | The data of the first ten variables is read into #100 to #109.<br>(After the data is normally read, #8539 is set to 99.) |
| G337 P15000 Q20 R99 ;⇒<br>IF[#8539 NE 99]GOTO888; | The data of twenty variables is read into #15000 to #15019.(After the data is normally read, #8539 is set to 99.) |
| G337 P16000 R99 ;⇒<br>IF[#8539 NE 0]GOTO888;<br><br>G331 ; | The remaining variable data is read into #16000 to #16009.<br>(After the data is normally read, #8539 is set to 0.) |

## 6.4.5 FANUC Floppy Cassette Control

File data on the FANUC FLOPPY CASSETTE, FA CARD or FANUC PROGRAM FILE Mate can be read, prepared and deleted, by an opening command and the instruction of control code ”G339”.

| | |
|---|---|
| Read file data | (G330) |
| Make file and write data | (G330) |
| Read directory information | (G330/G339 P1) |
| Delete file | (G330/G339 P2) |
| Change file data | (G330/G339 P3) |

(1) Reading File Data (G330)

G330 Pp Bb Ss Cc Ll/Ff/Aa ;

Specify one of the address ”L/F/A” at the time when the line is open by read control mode. At the result of this action, the head of specified file and file data reading is conducted. Refer to Section 6.4.2(1) ”Circuit open” about address ”P/B/S/C”. However, specify reading control (11/12) for address ”P” and ISO(2) for address (C).

Select one of addresses L,F, or A depending on the process.

(a) Head call by file name

Head call by file name is possible if the head variable number of the variable string which is housed in the file name is specified by address”L”.
Set the file name by decimalized ASCII code to the variable string of 17 variables, and specify the head variable number by address ”L”.

(Example)

When calling the file name ”ABCD”, set 65(A), 66(B), 67(C), 68(D), 32...32(space) to 17 variables of common variables #100-116.

G330 P11 B10 S2 C2 L100 ;

**NOTE**
1 A file name is fixed to 17 characters. Be sure to use 17 characters.
2 Specify the file name with alphanumerics and space. Alarm 8 will be generated if code 32(space) is specified at the head of file name.

(b) Head call by the file number

Head call can be conducted with a file number, if the file number (1-9999) is specified with address ”F”.

(Example)

When making a head call of file number 3

G330 P11 B10 S2 C2 F3 ;

(c) Head call of next file

Next file’s head call, which already exists can be conducted by the specification of address ”A”. Use this function to read files continuously. Be sure to specify ”A1” for address A”. End code (=8) is the other specification.

(Example)

When making a head call of the next file which is being called now

G330 P11 B10 S2 C2 A1 ;

(2) Making Files and Writing Data (G330)

G330 Pp$B_b$ Ss Cc (Ll/Ff) ;

New files are made and data can be written on a FANUC cassette by specifying one of address L or F when the communication line is opened with transmission control mode.

Refer to 6.4.2(1) ”Circuit open” for the specification of address ”P/B/S/C”. Be sure to specify transmission control (21/22) for address ”P”, and ISO (2) for address ”C”.

Select one of address L or F by processing.

(a) Making files by file name

A new file can be made and data can be written on a FANUC cassette with a specified file name, if you specify the head variable number of a variable string describing a file name by address ”L”.
Set the file name by decimalized ASCII code to a variable string of 17 variables. Set the head variable number with address ”L”.

(Example)

If you make a file of file name ”ABCD”;

65(A), 66(B), 67(C), 68(D), 32...32(space) shall be set to the 17 common variables of #100-116

G330 P21 B10 S2 C2 L100 ;

**NOTE**
1 A file name is fixed to 17 characters. Be sure to fill spaces(code 32), if the file name is less than 17 characters.
2 Specify a file name with alphanumeric and spaces. Alarm 8 will be generated if the 32(space) code is specifyed at the head of the file name.
3 A new file is added to the end of recorded files.

(b) Making files by file number

A new file can be made and data can be written to a specified file number by specifying the file number (1-9999) after address ”F”.

(Example)

To make file No.3, specify as follows:

G330 P21 B10 S2 C2 F3 ;

**NOTE**
1 If you make a file with the file number, be careful that every file which was previously input in the specifyed file number, and all subsequent file numbers are deleted.
2 Preparation of files by file number can only be conducted for the existing files. Files can be added only by file name specification.

(3) Control of Directory Information (G330/G339)

The control of directory information is possible with control code ”G339”.

Read directory information (G339 P1)

| | |
|---|---|
| Delete file | (G339 P2) |
| Change file name | (G339 P3) |

The control mode at line opening shall be of directory information to use this function. Instruct P=31/32 at the opening time for specification of directory information.

G330 Pp Bb Ss Cc ;

P31 : Channel 1 Directory information control

P32 : Channel 2 Directory information control

Refer to the section on line opening for address ”B/S/C”. Be sure to specify ISO (2) for address ”C”.

Do not instruct several functions continuously in one opening time for every function of directory reading, file deleting, file name changing if you open with the control mode of directory information.

For instance, close the line once to delete the file after reading and checking directory information, as follows;

1. Open with the control mode of directory information
2. Read directory information
3. Close the line
4. Open with the control mode of directory information
5. Delete a file
6. Close the line

The same function can be instructed several times in one openning.

(a) Reading directory information

G339 P1 Ff Ll Ss ;

File directory information can be read to a specified variable by specifying ”G339 P1”.

Specify the file number(1-9999) for which the directory information is to be read by address ”F”.

Address ”L” specifies the head variable number of a continuous 17 variables which houses the read file name. The file name is read with decimalized ASCII code every character.

Address ”S” specifies the variable number which houses the read file size.

If you have specified (F specification) the file number once, the next directory information can be read by instructing ”G339 P1” without F specification.

End code =11 reports that the directory of the specified file number does not exist.

(b) Deleting a file (G339 P2)

G339 P2 Ll/Ff ;

The relevant file can be deleted by the specification of ”G339 P2”.

Specify the head variable number of the variable string which houses the file name (ASCII code) you wish to delete by address ”L”, or specify the file number of the file you want to delete by address ”F”.

Be carefull when deleting with the file number because subsequent file numbers will change after deletion.

(c) Changing the file name (G339 P3)

G339 P3 Ff Ll ;

The recorded file name can be changed by the specification of ”G339 P3”.

Specify the file number (1-9999) of the file name you wish to change, by address ”F”.

Specify the head variable number of the variable string that houses the file name (ASCII code) to be changed, with address ”L”.

## 6.4.6 End Code (#8539)

Check the end code when G330-G339, has been instructed. There is no end code for line closing processing. It is always a normal end.

| #8539 | Contents |
|---|---|
| 0 | Normal end |
| 1 | Line doesn’t open |
| 2 | Line trouble ,DSR signal OFF |
| 3 | Line trouble, Over run |
| 4 | Line trouble, Receive buffer over flow |
| 5 | Line trouble, Framing error, parity error |
| 6 | No option for line function |
| 7 | Line use |
| 8 | Error in parameter<br>G335 instructed with transmit control mode<br>G336 instructed with read control mode |
| 9 | Data format error |
| 10 | Error on file number |
| 11 | File of file number specified to read directory information does not exist |
| 115 | Undefined variable number specified |
| 255 | Receive data does not exist |

# 6.5 REFERENCING AND READING NC PROGRAM WITH CONVERSATIONAL MACRO

## 6.5.1 Outline

Recording, deleting and changing of NC part program is possible by the conversational macro function. Control the NC program with program number and block number. The block numbers are counted on each EOB regarding the block with address ”O” being the No.1 block. For NC programs in the converational macro , one word is expressed by 2 variables of an address code and a number, and several words construct a block.

Set extension function valid (Parameter EXT1(No. 9002#7)) to ”1” at compiling when using this function.

Make the control instruction using G codes (G320-G329) in the conversational macro.

There is an end code #8529 to check that the instructed function run correcctly or not. Check end code after running G320-G329.

The end is normal when the end code #8529 is 0, and in the cases other than 0, the same number as the P/S alarm in the usual editing process will be notified.

Control variable

#8520 . . . . Specification of program number

#8521 . . . . Specification of block number

#8522 . . . . Specification of housed variable number

#8523 . . . . Variable number of a variable that specifies the number of digits under a decimal point.

#8529 . . . . End code(for reading only)

Control codes

G320 . . . . Record a new block

G321 . . . . Delete block

G322 . . . . Condensation of program

G325 . . . . Read specified block

G326 . . . . Write block

G327 . . . . Delete block

G328 . . . . Read a specified block coded in characters

G329 . . . . Write a specified block coded in characters

Background editing function as NC function is necessary for this function.

## 6.5.2 Recording of a New Program

G320 ;

Specify the program number to #8520 and instruct G320 to record a new program.

(Example)

To record O0002,

#8520=2 ;
G320 ;
IF[#8529 NE 0] GOTO 900 ;
Recording end
N900 ;
Error

Conduct similar processing as the edit ”Oxxxx”+”INSERT” when recording a new program. EOB is not inserted.

O0002

%

## 6.5.3 Deleting a Program

G321 ;

Specify the program number and instruct G321 to delete a program.

(Example)

To delete O0003

#8520=3 ;

G321 ;

IF[#8529 NE 0] GOTO900 ;

End of deletion

N900 ;

Error

## 6.5.4 Reading a Specified Block

(1) Reading a specified block coded in words (G325)

When issuing this command, specify a program number, block number, and the number of the variable into which the block is to be read. From the corresponding NC program coded in words, the corresponding block is read into the specified variable area. Each word contains two variables: an address code and numerical value.

(Example)

O0004 ;

G92 X0. M08 ;

G90 G00 X10.5 M05 ;

#8520=4 ;

#8521=3 ;

#8522=100 ;

G325 ;

IF[#8529 NE 0] GOTO900 ;

End of reading

N900 ;

Error

The following program data is loaded from #100 which is specified by variable number #8522, when the above instruction is executed.;

| | |
|---|---|
| #100=7 | Address ”G” |
| #101=90 | Number |
| #102=7 | Address ”G” |
| #103=0 | Number |
| #104=24 | Address ”X” |
| #105=10.5 | Number |
| #106=13 | Address ”M” |
| #107=5 | Numbew |
| #108=27 | Address ”EOB” |

EOR (28) is housed as an address, if EOB is not at the end block of the program or EOR position is specified with block number.

Reading is not conducted as the end code becomes ”255”, if a block No. exceeding the block of EOR is specified.

(Example)

O0004 ;

G92 X0. M08 ;

M02

%

| | |
|---|---|
| #100=13 | Address ”M” |
| #101=2 | Number |
| #102=28 | Address ”EOR” |

(Example)

If

O0004 ;

G92 X0. M08 ;

%

#100=28 Address ”EOR”

If the function for reading a specified block coded in words (G325) reads a block which is not coded in words, completion code #8529 is set to 253 to indicate the fact.

When the completion code is set to 253, use the function for reading a specified block coded in characters (G328) to read the same block.

#8520 = program-number ;

#8521 = block-number ;

#8522 = number-of-the-variable-into-which-the-block-is-read ;

G325 ;

IF[#8529 EQ 253]GOTO100 ; Conventional processing

N100 G328 ;

Character-type analysis

(2) Reading a specified block coded in characters (G328)

When entering this command, specify a program number and block number. From an NC program which is not coded in words, the corresponding block can be read in units of characters into a specified variable area. The block is read in the form of decimal ASCII codes. Each control command (WHILE/IF/...) or function (SIN/COS/FUP/...) is represented as a single data item consisting of special code.

#8520 = program-number ;

#8521 = block-number ;

#8522 = 100 ;
(Number of the variable into which the block is read)

G328 ;

IF[#8529 NE 0]GOTO900 ; ← Error check

When the corresponding block is ”#1 = SIN [#2];”

#100 : 35 (23H) ”#”
#101 : 49 (31H) ”1” The data is input as shown on the left.
#102 : 61 (3Dh) ”=”
#103 : 276 (114H) ”SIN”
#104 : 91 (5BH) ”[”
#105 : 35 (23H) ”#”
#106 : 50 (32H) ”2”
#107 : 93 (5DH) ”]”
#108 : 59 (3BH) ”;” ← ”EOB” code

## 6.5.5 Block Writing

(a) Writing a block coded in words (G326Pp)

Prepared program data can be written in the variable area after the block specified by the program number and block number. Specify the maximum number of variable data by address P. If address ”EOB” is present within the specified data, program is written up to ”EOB”; if ”EOR” is present, program is written up to the EOR; and if neither ”EOB” nor ”EOR” is present, program is written by the number of data specified by address ”P”.

(Example)

O0004 ;

G92 X0. M08 ;

G90 G00 X10.5 M05 ;

#8520=4 ;

#8521=2 ;

#8522=100 ;

#100=7 ;

```
#101=1 ;
#102=24 ;
#103=20.5 ;
#104=6 ;
#105=1000 ;
#106=20.7 ;
G326 P7 ;
IF[#8529 NE 0] GOTO900 ;
End of writing
N900 ;
Error
```

If the above instruction is executed, a block is input as follows;

```
O0004 ;
G92 X0. M08 ;
G1 X20.5 F1000. ;
G90 GO X10.5 M05 ;
```

End code becomes ”255” and it is impossible to conduct writing, if the specification of block number is EOR block only or higher numbers.

(1) Decimal point specification of every address in block writing

The number of digits after the decimal point of every address can be specified. With this specification, the contents of the variable number which was specified to #8523 becomes the number of digits after the decimal point of address A, and it is possible to determine the number of digits after the decimal point of every address, as shown below.

#8523=501 ;

#501 is the number of digits after the decimal point of address A

#502 is the number of digits after the decimal point of address B

:

:

#525 is the number of digits after the decimal point of address Y

#526 is the number of digits after the decimal point of address Z

Specify <vacant> or 0 - 7 for the number of digits after the decimal point. In the case of <vacant>, it is considered that no decimal point exists in the address.

(Example)

If address code=A, number=1.2345678

Decimal point specification

| | | |
|---|---|---|
| = <blank> | A1 | |
| = 0 | A1. | |
| = 1 | A1.2 | |
| = 2 | A1.23 | |
| = 3 | A1.235 | * |
| = 4 | A1.2346 | * |
| = 5 | A1.23457 | * |
| = 6 | A1.234568 | * |
| = 7 | A1.2345678 | |

* The data less than the specified digits is rounded off.

When #8523 is 0, auto setting is conducted by the system conditions (mm input/inch input, 0.1µm instruction) and specified address. #8520 - #8523 is set to 0 when the power is turned on.

(2) Writing a block coded in characters (G329Pp)

Use this command to write a program whose data is not coded in words. This command writes the program data coded in units of characters in a variable area.

The block to be written must be defined beforehand in ASCII code in a macro variable area. This command writes the data after the block specified by the program number and block number.

The maximum number of variable data items is indicated by address P. If EOB or EOR is detected while the specified variable data is being written, data writing ends at the EOB or EOR. When neither EOB nor EOR is found, the complete data indicated by address P is written.

(ASCII string definition)

#8520 = program-number

#8521 = block-number

#8522 = number-of-variable-at-the-beginning-of-the-ASCII-string

G329 P20 ;

IF[#8529 NE 0]GOTO900 ; ← Error check

> **NOTE**
> This command cannot catalog a program. If an attempt is made to catalog a program with this command (if the data to be written begins with O), an error code is indicated and #8529 is set to 202.

## 6.5.6 Block Deletion

G327 ;

A block specified by a program number and a block number can be deleted.

#8520=4 ;

#8521=3 ;

G327 ;

IF[#8529 NE 0] GOTO900 ;

End of deletion

N900 ;

Error

The block of block No.3 of program O0004 is deleted by the above instruction.

### 6.5.7 Condensation of Program

When writing an NC program prepared by a conversational macro onto the NC's tape memory, the program can be condensed to extend memory area.

Set a program number of a program to be condensed to the variable #8520 and specify G322. Then the program is condensed.

The result of condensation is informed to the end code #8529.

(Example) To condense O1234

```
#8520=1234;
G322;
IF [#8529 NE 0] GOTO Error;
```

### 6.5.8 End Code (#8529)

Check the end code after running every processing.

The end is normal when the end code is 0, and when the end code is a number other than 0, the same number as with the P/S alarm in the usual edition processing and the following codes are informed.

| #8529 | Contents |
|---|---|
| 140 | Processing of the program being selected in the foreground cannot be conducted. |
| 200 | The specified character code is not found. (G329) |
| 202 | The data to be written begins with O. (G329) |
| 253 | The specified block is not coded in words. |
| 254 | Program editing is prohibited by setting the memory protection key [when compilation parameter bit 1 of No. 9006 (KEYC) is set to 0]. |
| 255 | Need the option for background editing function.<br>It is impossible to instruct the statement during background editing.<br>Parameter for the extension function effective (No.9002#7 EXT1) must be set to "1" at compiling. |

### 6.5.9 Notes

Selection processing by work number search etc. is required when foreground operation is executed for a program which has been prepared and edited under background.

### 6.5.10 Address Code Table

| Address | Code | Address | Code | Address | Code |
|---|---|---|---|---|---|
| A | 1 | B | 2 | C | 3 |
| D | 4 | E | 5 | F | 6 |
| G | 7 | H | 8 | I | 9 |
| J | 10 | K | 11 | L | 12 |
| M | 13 | N | 14 | O | 15 |
| P | 16 | Q | 17 | R | 18 |
| S | 19 | T | 20 | U | 21 |
| V | 22 | W | 23 | X | 24 |
| Y | 25 | Z | 26 | | |
| EOB | 27 | EOR | 28 | / | 29 |

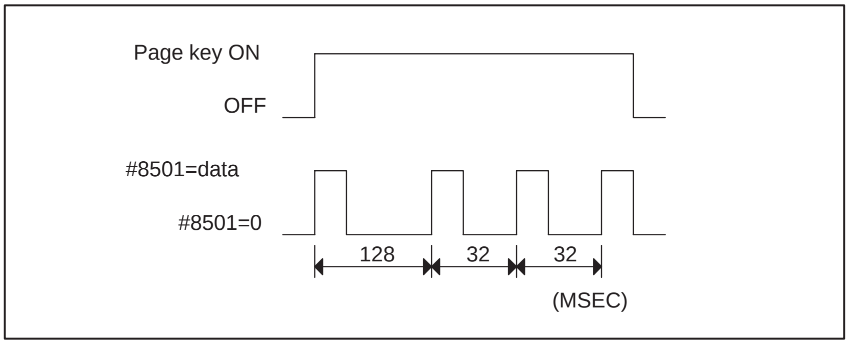
| Statement | Decimal | Hexadecimal | Statement | Decimal | Hexadecimal |
|---|---|---|---|---|---|
| IF | 258 | 102H | SIN | 276 | 114H |
| WHILE | 259 | 103H | COS | 277 | 115H |
| GOTO | 260 | 104H | TAN | 278 | 116H |
| DO | 261 | 105H | ATAN | 279 | 117H |
| END | 262 | 106H | SQRT | 280 | 118H |
| GE | 264 | 108H | ABS | 281 | 119H |
| GT | 265 | 109H | BCD | 282 | 11AH |
| LE | 266 | 10AH | BIN | 283 | 11BH |
| LT | 267 | 10BH | FIX | 284 | 11CH |
| NE | 268 | 10CH | FUP | 285 | 11DH |
| EQ | 269 | 10DH | ROUND | 286 | 11EH |
| XOR | 272 | 110H | POPEN | 291 | 123H |
| OR | 273 | 111H | PCLOS | 292 | 124H |
| AND | 274 | 112H | DPRNT | 293 | 125H |
| | | | BPRINT | 294 | 126H |
| | | | SETVN | 295 | 127H |

## 6.6 CONTINUOUS INPUT BY CURSOR AND PAGE KEY

Continuous input by cursor and page key is possible with the conversational macro key input control (#8501).

(Address code table)

#8501 can be read with the following timing if the cursor and page key, are continuously pressed.




**NOTE**
The value of #8501 remains as is until read by a macro statement. Once the value has been read, all key entries are ignored.

## 6.7 MASKING OF O, N NUMBER APPEARANCE

O, N number appearance can be erased on the CUSTOM screen.

O, N numbers does not appear when compile parameter ”ONMSK=1” (No.9003#0).

## 6.8 READING AND PRESETTING CUTTING TIME AND CUTTING DISTANCE BY CONVERSATIONAL MACRO

Cutting distance and cutting time can be read and preset with the macro variable of the conversational macro.

Use it for the control of a tool life.

#8553, #8554 does not bocome 0 at power on.

### 6.8.1 Reading and Presetting Cutting Time (#8553)

The time only for instruction of G01 (linear interpolation) and G02,G03 (circular interpolation) can be counted by #8553.

The unit is the same as the macro variable (#3002).

### 6.8.2 Reading and Presetting Cutting Distance (#8554)

The distance only for instruction of G01(linear interpolation), and G02, G03 (circular interpolation) can be counted by #8554.

It is valid also for helical instruction.

Parameters (No.9002#7 / EXT1) and (No.9004#7 / CUTLG) have to be on at compiling to use this function.

The unit is

... 0.001(mm) for the mm input system.

... 0.0001(inch) for the inch input system.

As the tool movement distance is added to #8554 at the time of starting the cutting block, the distance of block movement is added even when cutting stops during a block due to resetting, etc.

## 6.9 READING AND PRESETTING RELATIVE COORDINATES BY CONVERSATIONAL MACRO

Relative coordinates can be read and preset with the conversational macro.

### 6.9.1 Reading Relative Coordinates

Reading of relative coordinates can be informed by the window function.

Set ID No.110-115 to #8993, and relative coordinates can be read by #8999.

ID No. 110 1st axis coordinates
111 2nd axis coordinates
112 3rd axis coordinates
113 4th axis coordinates
114 5th axis coordinates
115 6th axis coordinates
116 7th axis coordinates
117 8th axis coordinates

(Example)

If the relative coordinates value of the 1st axis is –123.456
#8998=110; ((Include the contents of ID No.110)
#500=#8999;
Contents of #500 is -123456.

### 6.9.2 Presetting Relative Coordinates

Relative coordinate values can be freely preset by G310AaQq.

Set the ID No. shown above to address A.

A110 - 115

Set the coordinates values to be preset to address Q.

Q –99999999 to 99999999

Relative coordinate values can be preset freely by execution of this control.

(Example)

To preset the relative coordinates value of 1st axis to -123.45;
G310 A110 Q-123450
Be sure that this function does not change the increment system.

## 6.10 KEY–INPUT AND DATA–INPUT CONTROL

(1) Key-input control variable #8501

You can read out the key input with #8501. The following is the correspondence of the key input and #8501.

Display unit with seven soft keys

| Key input | Value | Key input | Value |
|---|---|---|---|
| PAGE.DOWN | 1 | SOFT FUNCTION KEY LEFT | 11 |
| PAGE UP | 2 | SOFT FUNCTION KEY 1 | 12 |
| CURSOR DOWN | 3 | SOFT FUNCTION KEY 2 | 13 |
| CURSOR UP | 4 | SOFT FUNCTION KEY 3 | 14 |
| ALTER | 5 | SOFT FUNCTION KEY 4 | 15 |
| INSERT | 6 | SOFT FUNCTION KEY 5 | 16 |
| DELETE | 7 | SOFT FUNCTION KEY RIGHT | 17 |
| INPUT | 8 | | |
| RESET | 10 | | |
| CURSOR RIGHT | 18 | | |
| CURSOR LEFT | 19 | | |

Display unit with twelve soft keys

The soft function keys follows:

The other keys are the same as seven softy keys.

| Key input | Value | Key input | Value |
|---|---|---|---|
| SOFT FUNCTION KEY LEFT | 20 | SOFT FUNCTION KEY 1 | 21 |
| SOFT FUNCTION KEY 2 | 22 | SOFT FUNCTION KEY 3 | 23 |
| SOFT FUNCTION KEY 4 | 24 | SOFT FUNCTION KEY 5 | 25 |
| SOFT FUNCTION KEY 6 | 26 | SOFT FUNCTION KEY 7 | 27 |
| SOFT FUNCTION KEY 8 | 28 | SOFT FUNCTION KEY 9 | 29 |
| SOFT FUNCTION KEY 10 | 30 | SOFT FUNCTION KEY RIGHT | 31 |

When there is no key input, the #8501 value is 0.

Once there is a key input, #8501 holds its value till it is input with the macro statement, and the following key inputs will be ignored. When #8501 is input, the status gets ready to accept the key input and #8501 will be 0.

It is impossible to write a value into #8501.

(2) Data input control variable #8502, numerical data variable #8503, address data variable #8504

Control the input of numeric data and address data by setting the following values to #8502.

#8502 = 0 ...No data input
= 1 ... Input of numeric data
= 2 ...Input of address data and numeric data
= 3... Expanded data input control

When #8502 is 0, nothing will be displayed on the data input line and it will be impossible to input data, either.

When #8502 is 1, NUM will be displayed on the data input line and input of numeric data will be possible. When #8502 is 2, ADRS will be displayed on the data input line. After address data is input, NUM is displayed and it is possible to input the numeric data. #8502 value is 0 at switching on the power.

When the key-input control variable #8501 is key-input to the value other than 0, the data input line will return to the initial state. It is possible to read out the input numeric value and address with the numeric data variable #8503 and the address data variable #8504 respectively. The following is the correspondence of the input address and #8504.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| A | ...1 | B | ...2 | C | ...3 | D | ...4 | E | ...5 |
| F | ...6 | G | ...7 | H | ...8 | I | ...9 | J | ..10 |
| K | ..11 | L | ...1 | M | ..13 | N | ..14 | O | ..15 |
| P | ..16 | Q | ..17 | R | ..18 | S | ..19 | T | ..20 |
| U | ..21 | V | ..22 | W | ..23 | X | ..24 | Y | ..25 |
| Z | ..26 | | | | | | | | |

#8503 and #8504 values will be held till the input control variable #8501 is key-input to the value other than 0.

When numerical data and address data are not input, #8503 and #8504 will be ”vacant” respectively.

It is impossible to write any values to #8503 and #8504.

<u>Conversational Macro Extented Data Input Control Function</u>

If you set 3 to #8502, and a variable number to #8552, the character string input mode is entered, and < apears on the input line, and it is possible to input addresses and number data.

The data input line changes to the initial state when key input status variable #8501 changes to other than 0. 32 variables from a variable number which is set to variable #8552 can read the input numbers and addressesfor as ASCII codes.

<blank> input a 32 number series if there is no input data.

(Example)

Set for

#8502=3 ;

#8552=500 ;

and input

0123456ABCD<

and press return key, then the following values are set to the variables.

#500=48, #501=49, #502=50, #503=51, #504=52, #505=53, #506=54, #507=65, #508=66, #509=67, #510=68, #511=<blank> to #531=<blank>, #8501=8

The values of #8503, #8504 are not guaranteed.

## 6.11 CURSOR CONTROL

It will be possible to display the cursor by setting the value at the cursor control variable #8505.

#8505 = 0 .. Cursor erase

= 1 .. Cursor display

#8505 value is 0 at power ON.

It will be possible to display the cursor at any optional position by setting the values for the cursor X position control variable #8506 and the cursor Y position control variable #8507. Specify the cursor position with the character coordinates system.

## 6.12 PROCESSING ARRAY TYPE P–CODE VARIABLES

(1) Reference of array type P-CODE variables for conversational macro

In conversational macro, two or three-dimensional array type P-CODE variables (#10000...) can be referred. Set a proper value to the following array control variables beforehand and an array element of P-CODE variables is referred to corresponding to the variable number (#1 - #99).

> **NOTE**
> Variables #1 to #99 for convesational macro are different from local variables #1 to #33 for execution macro.
> Array control variable:
> #8512 Two-dimensional array number
> #8513 Three-dimensional array number
> #8516 Maximum value of one-dimensional array number
> #8517 Maximum value of two-dimensional array number
> #8519 Array top variable number

Specify an array type by the array control variables #8516 to #8519 and specify the array number referred to by the #8512 and #8513. Variables #1 to #99 are used to refer the values.

The correspondence between the P-CODE variables and array elements is as shown below:

P-CODE variable number

= #8519 + ((#8516 × #8517) × (#8513 × 1)) + (#8516 × (#8512–1)) (Specified variable number–1)

Example)

When the #8516, #8517, and #8519 are set to 10, 5, and 10100, respectively:

1) When both of #8512 and #8513 are set to 1, the value for #1 corresponds to that for #10100.

2) When the #8512 and #8513 are set to 3 and 2, respectively, the value for #10 corresponds to that for #10179.

Each variable of #8512 to #8517 and #8519 are set to 1 and 10000, respectively when the power is turned on.

Thus, the #8513 and #8517 can be used in the case of use as the two-dimensional array. No check is carried out on minimum value of each variable P-CODE variable number after calculation.

Use the macro program when needed.

> **NOTE**
> This function can be used in the conversation macro program and auxiliary macro program.
> In the execution macro, the #1 to #33 are local variables and the #34 to #99 are variables which cannot be used and would result in an alarm.

(2) Processing array type P-CODE variables

This is a function which controls processing of array type variable or of variable string when P-CODE variables are used in the conversational macro.

1) clearing array type number, variable string (continuous writing of specified data)

2) transferring to variable string from array type variables or variable string

Every processing commands control code ”G315” after defining an array or variable string or data to the following control numbers.

| | | |
|---|---|---|
| #8511 | : | data to be transferred |
| #8512 | : | 2 dimensional array number or the head variable of a variables string to be transferred |
| #8513 | : | 3 dimensional array number to be transferred |
| #8514 | : | transferred 2 dimensional array number or the head variable of a variable string |
| #8515 | : | transferred 3 dimensional array number |

Control code

G315 P (Processing code) K (a number of processing data) ;

Processing code specifies the type of processing with a 3-digit number. High-order ”0s” can be supressed.

| | | |
|---|---|---|
| P001 (P1) | : | Loads data of #8511 to K continuous numbers from the number which was specified at #8514. |
| P002 (P2) | : | Transfers K continuous data which was specified at #8512 to K continuous variables from the variable which was specified at #8514 (ascending order). |
| P003 (P3) | : | Transfers K continuous data which was specified at #8512 to K continuous numbers from the number which was specified at #8514 (descending order). |
| P101 | : | Loads data of #8511 to K continuous array numbers of array number #1 which was specified at #8514, #8515. |
| P102 | : | Transfers K continuous array variables starting with variable #1 which was specified at #8512 and #8513, to the continuous array variable which was specified at #8514 and #8515 (ascending order). |
| P103 | : | Transfers K continuous array ariables starting with variable #1 which was specified at #8512 and #8513, to the continuous array variable which was specified at #8514 and #8515 (dscending order). |

The difference of P2 and P3, or P102 and P103 results from whether transfer processing is conducted from a small number or from a large number.

(Example)

If #8512=10000, #8514=10010,

G315 P2 K3 ; has the following disposition.

#10010=#10000 ;

#10011=#10001 ;

```
    #10012=#10002 ;
G315 P3 K3 ; has the following disposition
    #10012=#10002 ;
    #10011=#10001 ;
    #10010=#10000 ;
```

## 6.13 TORQUE LIMIT CONTROL

The torque limit override can be modified to the specified value by setting values to the #8621 to #8628.

| | | | |
|---|---|---|---|
| #8621 | 1st axis | #8625 | 5th axis |
| #8622 | 2nd axis | #8626 | 6th axis |
| #8623 | 3rd axis | #8627 | 7th axis |
| #8624 | 4th axis | #8628 | 8th axis |

The relation of setting value and torque limit override are shown as below:

| Setting | Torque override |
|---|---|
| 0 | 0% |
| : | : |
| 128 | 50% |
| 255 | 100% |

The values on power on are 100%.

**NOTE**
When a value other 0 to 255 is set:
In an execution---P/S alarm 110
In conversational/auxiliary macro---Ignored

## 6.14 DATA READING OF A/D CONVERTER (Series 16/18)

Data from the A/D converter can be read by the conversational macro.

The A/D converted data from channels 1 to 4 corresponds to macro variables #8631 to #8634.

When reading operation is performed in the conversational macro program, the A/D converted data of each channel at that timing is input.

#8631 A/D converted data of channel 1

#8632 A/D converted data of channel 2

#8633 A/D converted data of channel 3

#8634 A/D converted data of channel 4

The input data is calculated in proportion to the following:

–10V = 0
0V = 128
+10V = 255

**NOTE**

1 The A/D converted data variables #8631 to #8634 can be used for read only.No data can be written in them.

```
Correct      #100 = #8631
             IF [#8631 GT 1000] GOTO 100 ;
             #101 = #8634/200 ;
Incorrect    #8631 = 2000 ;
```

2 The PCB ”Option 2 board (analog I/O)” (A16B - 1211 - 0960) is required when this function is used.

## 6.15 KEY–IN LINE CONTROL

On the conversational macro screen, a key-in line is displayed at a point whose X and Y coordinates are 0 and 20 respectively. Up to 32 characters (20 lines when CNC parameter no. 3105#2 = 1) can be input. The prompt and key-in lines are displayed in light blue. The conversational macro can control the display position, number, and color. When the screen is switched from the POS screen to the conversational macro screen, the setting is initialized. When switching the screen, create a P-CODE program to set control variables.

(1) #8561, #8562 (X and Y coordinates of the point where the key-in line is displayed)

#8561 : X coordinate

#8562 : Y coordinate

The display position is changed when #8562 is written.

The macro program must be created so that #8561 and #8562 are set in that order. The previous key-in line is not automatically erased after the display position is changed. If required, erase the previous key-in line with a macro routine.

(2) #8563 (Number of input keys)

#8563 : Number of input keys

Previous key-in lines are not automatically erased after the number of input keys is reduced. If required, erase the previous key-in lines with a macro routine.

(3) #8564, #8565 (Display of prompt and key-in line)

To change the display of the prompt, set #8564 as listed below.

To change the display of the key-in line, set #8565 as listed below. The set values are listed below.

| | Standard | Blinking | Reversed | Blinking and reversed |
|---|---|---|---|---|
| White | 224 | 232 | 240 | 250 |
| Light blue | 192 | 200 | 208 | 216 |
| Purple | 160 | 168 | 176 | 184 |
| Blue | 128 | 136 | 144 | 152 |
| Yellow | 96 | 104 | 112 | 120 |
| Green | 64 | 76 | 80 | 88 |
| Red | 32 | 40 | 48 | 56 |

## 6.16 READING THE BACKGROUND EDITING STATUS

The conversational and auxiliary macros can read the CNC background editing status with a variable.

#8526: Background editing status
0 : The background editing is halted.
1 : The background editing is in progress.

> **NOTE**
> Variable #8526 cannot be written.

## 6.17 READING THE NUMBER OF CATALOGED PROGRAMS

The conversational and auxiliary macros can read the number of programs cataloged in the CNC program memory with a variable.

#8527 : Number of cataloged programs

> **NOTE**
> Variable #8527 cannot be written.

## 6.18 READING THE SIZE OF THE FREE SPACE IN THE CNC PROGRAM MEMORY

The conversational and auxiliary macros can read the size of the free space remained in the CNC program memory with a variable.

#8528: Size of free space in CNC program memory
(Unit: Characters)

> **NOTE**
> Variables #8528 cannot be written.

## 6.19 READING THE REMAINING TRAVELING DISTANCE

The conversational and auxiliary macros can read the remaining traveling distance of each CNC control axis with a variable.

#5181: Remaining traveling distance of the first axis
#5182: Remaining traveling distance of the second axis
: :
#5188: Remaining traveling distance of the eighth axis

> **NOTE**
> Variable #5181 to #5188 cannot be written.

## 6.20 OFFSET MEMORY C AND EXTENDED SYSTEM VARIABLE OF THE WORKPIECE COORDINATE SYSTEM (Series 16/18)

When the execution, conversational, or auxiliary macro references offset memory C or an extended system variable of the workpiece coordinate system, 100000 is added to the corresponding variable number used by the custom macro function of the system.

(Example) Reading and writing the compensation data of the offset memory (C) by the execution, conversational, or auxiliary macro

#100 = #110001;
#110002 = 1.5;

# 6.21 PMC AXIS CONTROL

**NOTE**
PMC axis control is not available for 20-TA/21-TA.

## 6.21.1 PMC Axis Control by the G Code

The conversational macro can control the PMC control axis through the PMC axis control interface. The seven control codes listed below are used for this purpose. The PMC control axis to be controlled is specified by the PMC control axis selection variable, #8602.

G340 → Rapid traverse command
G341 → Cutting feed command
G344 → Dwell command
G345 → Reference position return command
G346 → Miscellaneous function command
G348 → Status signal read command
G349 → Command signal write command
#8602 → PMC control axis selection variable

| #8602 | Area | Series 16 | Series 18 | 20–FA | Series 21 |
|---|---|---|---|---|---|
| 0 | Area A | 1st to 8th axis | 1st to 4th axis | 1st to 3rd axis | 1st to 4th axis |
| 1 | Area B | 1st to 8th axis | 1st to 4th axis | 1st to 3rd axis | 1st to 4th axis |
| 2 | Area C | 1st to 8th axis | 1st to 4th axis | 1st to 3rd axis | 1st to 4th axis |
| 3 | Area D | 1st to 8th axis | 1st to 4th axis | 1st to 3rd axis | 1st to 4th axis |

**NOTE**
1 For information about the PMC axis control interface, refer to the corresponding connection manual for each machine.
2 A control axis is selected by setting bits of the input signal parameter G136. Use the PMC to set the bits of G136.

For details, refer to the CONNECTION MANUAL for each Series.

If a value other than 0 to 3 is specified in #8602, the control command is ignored.

(1) Control codes

(a) Rapid traverse command (G340)

G340 Xxxx;

The rapid traverse command is issued to the PMC control axis. Specify the incremental traveling distance after address X.

(b) Cutting feed command (G341)

G341 Xxxx Ffff;

The cutting feed command is issued to the PMC control axis. Specify the incremental traveling distance after address X. Specify the feedrate after address F.

(c) Dwell command (G344)

G344 Pxxx:

The dwell command is issued to the PMC control axis. Specify the dwell time after address P.

(d) Reference position return command (G345)

G345;

The reference position return command is issued to the PMC control axis.

(e) Miscellaneous function command (G346)

G346 Mmm;

The miscellaneous function command is issued to the PC control interface. Specify the miscellaneous function code after address M.

(f) Status signal read command (G348)

G348 Pppp;

The status signals (F130, F133, F136, F139) of the corresponding PMC axis control interface are read into the variable indicated by address P. A one-byte signal is expressed in decimal and input to the variable. The EBSY signal is always input as 0.

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| EBSY | EOTN | EOTP | EGEN | EDEN | EIAL | ECKZ | EINP |

For details of the signals, refer to the CONNECTION MANUAL for each Series.

(Example) When both EOTN and EIAL are 1
G348 P100;
This G code causes #100 to be set to 68.

(g) Command signal write command (G349)

G349 Pppp;

A numeric value indicated by address P is written into the command signals (G142, G154, G166, G178) of the corresponding PMC axis control interface. The EBUF and EFIN signals cannot be written.

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| EBUF | ECLR | ESTP | ESOF | ESBK | | | EFIN |

For details of the signals, refer to the CONNECTION MANUAL for each Series.

(Example) ECLR can be set to 1 by the following G code:
G349 P64; (64 = 01000000b)

(2) Notes

(a) Buffering a command

The PMC axis control function buffers command blocks in the CNC unit so that it can execute two or more commands sequentially. While a block is being executed, the next block can be specified if the CNC buffer has sufficient free space. If the CNC buffer has no free space, the next command is in the wait state until the previous command block is executed and space in the buffer becomes free. Meanwhile, the display level is locked.

(b) Miscellaneous function command

A miscellaneous function command can be issued by G346. The conversational macro, however, cannot control the miscellaneous function completion signal, EFIN. Use the PMC to control it.

(c) Contention for the PMC axis control function

The situation in which both the PMC and this function issue commands to an identical PMC control axis must be avoided. When this function is used for a PMC control axis, the PMC can control only the following two signals for the same control axis: Miscellaneous function completion signal (EFIN) and axis selection signal (G136).

For details, refer to the CONNECTION MANUAL for each Series.

## 6.21.2 PMC axis Control by Variables

The conversational macro can control the PMC control axis with variables through the PMC axis control interface.

The macro uses the following variables for PMC axis control:

#8700 → PMC control axis selection variable

| Name of variable | Variable area | | | |
|---|---|---|---|---|
| | Area A | Area B | Area C | Area D |
| PMC command signal variable | #8710 | #8720 | #8730 | #8740 |
| PMC control command variable | #8711 | #8721 | #8731 | #8741 |
| PMC cutting feedrate variable | #8712 | #8722 | #8732 | #8742 |
| PMC controlled travelling distance variable | #8713 | #8723 | #8733 | #8743 |
| PMC status signal read variable | #8715 | #8725 | #8735 | #8745 |

| Variable | Series 16 | Series 18 | 20-FA | Series 21 |
|---|---|---|---|---|
| Area A | 1st to 8th axis | 1st to 4th axis | 1st to 3rd axis | 1st to 4th axis |
| Area B | 1st to 8th axis | 1st to 4th axis | 1st to 3rd axis | 1st to 4th axis |
| Area C | 1st to 8th axis | 1st to 4th axis | 1st to 3rd axis | 1st to 4th axis |
| Area D | 1st to 8th axis | 1st to 4th axis | 1st to 3rd axis | 1st to 4th axis |

**NOTE**
A control axis is selected by the PMC control axis selection variable, #8700.

For details, refer to the CONNECTION MANUAL for each Series.

(1) Variables

(a) PMC command signal variable
(#8710, #8720, #8730 and #8740)

When a numeric value is specified in #8710 or #8720, data is written into the command signals (G142, G154, G166, G178) of the corresponding PMC axis control interface. G142, G154, G166, and G178 cannot be read from the PMC. The EFIN signal cannot be written.

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| EBUF | ECLR | ESTP | ESOF | ESBK | | | EFIN |

(b) Control command variable
(#8711, #8721, #8731 and #8741)

When a control command is specified in #8711 or #8721, the axis control command is written into the corresponding axis control command signal (G143, G155, G167, G179). (G143, G155, G167, and G179 cannot be read from the PMC). The command can also be read.

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| | EC6 | EC5 | EC4 | EC3 | EC2 | EC1 | EC0 |

(c) Cutting feedrate control variable
(#8712, #8722, #8732 and #8742)

When a numeric value is specified in #8712 or #8722, the cutting feedrate is written into the corresponding command data signal (G144, G145, G156, G157, G168, G169, G180, G181). The feedrate can also be read. (G144, G145, G156, G157, G168, G169, G180 and G181 cannot be read from the PMC).

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| EIF7 | EIF6 | EIF5 | EIF4 | EIF3 | EIF2 | EIF1 | EIF0 |
| EIF15 | EIF14 | EIF13 | EIF12 | EIF11 | EIF10 | EIF9 | EIF8 |

(d) Controlled traveling distance variable
(#8713, #8723, #8733 and #8743)

When a numeric value is specified in #8713 or #8723, the axis traveling distance, dwell time, or miscellaneous function code is written into the corresponding command data signal (G146 to G149, G158 to G161, G170 to G173, G182 to G185). The data can also be read. (G146 to G147, G158 to G161, G170 to G173 and G182 to G185 cannot be read from the PMC).

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| EID7 | EID6 | EID5 | EID4 | EID3 | EID2 | EID1 | EID0 |
| EIF15 | EID14 | EID13 | EID12 | EID11 | EID10 | EID9 | EID8 |
| EID23 | EID22 | EID21 | EID20 | EID19 | EID18 | EID17 | EID16 |
| EIF31 | EID30 | EID29 | EID28 | EID27 | EID26 | EID25 | EID24 |

(e) PMC status signal read variable
(#8715, #8725, #8735 and #8745)

The status signals (F130, F133, G136, G139) of the corresponding PMC axis control interface are written into the variable specified by #8715 or #8725. A one-byte signal is expressed in decimal and input into the variable.

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| EBSY | EOTN | EOTP | EGEN | EDEN | EIAL | ECKZ | EINP |

For details of the signals, refer to the CONNECTION MANUAL for each Series.

## 6.22 INTERLOCK FUNCTION FOR AXIS DIRECTION

The interlock control variable for a single axis direction, #8600, determines the direction for which interlocks are provided. Variable #8601 indicates the axis and direction of a movement when the SKIP signal is set on.

This function is validated when compilation parameter XDIL (No. 9002, #4) is set to 1.

> **NOTE**
> 1 This function cannot be used with the Series 20.
> 2 This function cannot be used when the manual linear or circular interpolation function is supported (Series 16 and 18).

The interlock function for a single axis direction is validated only when the following two conditions are satisfied: The system is in the mode in which an interlock is provided for a single axis direction, that is, the JOG or HNDL mode. The signal of the internal PMC relay (R area) which was specified by compilation parameters 9035 and 9036 is set on.

When the values of variables #8600 and #8601 are expressed in binary, each digit corresponds to a single axis direction as shown below:

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| M Series | 4– | 4+ | Z– | Z+ | Y– | Y+ | X– | X+ |
| T Series | 4– | 4+ | 3– | 3+ | Z– | Z+ | X– | X+ |

When the values of #8600 and #8601 are 00000001 in binary, for example, they indicate the positive direction along the X-axis. When they are 00000010 in binary, they indicate the negative direction along the X-axis.

(1) Interlock control variable for a single axis direction, #8600

When the value of variable #8600 is 0, interlocks are provided for all axes whenever the SKIP signal is set on.

When the value of variable #8600 is other than 0, interlocks are provided for the axis direction indicated by the value.

When the power is turned on, the value of #8600 is set to 0.

(2) Variable indicating the axis and direction of movement when the SKIP signal is set on, #8601

When the state of the SKIP signal changes from off to on, variable #8601 indicates the axis along which the last movement was made and the direction of movement.

The value of #8601 is retained until the state of the SKIP signal changes from off to on next time.

No value can be written in variable #8601.

## 6.23 FUNCTION FOR SEPARATING UI FROM UO OF THE P–CODE PROGRAM

This function allows the signals of the system variables of the P-CODE program (UI: #1000 to #1015, #1032, UO: #1100 to #1115, #1132) to be different from those of the user program.

When the program is compiled with parameter DIOC (No. 9006, #0) set to 1, UI and UO of the P-CODE program (execution, conversational, or auxiliary macro) correspond to the following interface signals. Even when this function is used, UI and UO of a user-created custom macro program correspond to ordinary interface signals.

| Input signal | Output signal |
|---|---|
| #1000 : EUI00 | #1100 : EUO00 |
| #1001 : EUI01 | #1101 : EUO01 |
| #1002 : EUI02 | #1102 : EUO02 |
| : | : |
| #1015 : EUI15 | #1115 : EUO15 |
| #1032 : EUI00-EUI15 | #1132 : EUO00-EUO15 |

PMC interface

Input signal

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| G082 | EUI07 | EUI06 | EUI05 | EUI04 | EUI03 | EUI02 | EUI01 | EUI00 |
| G083 | EUI15 | EUI14 | EUI13 | EUI12 | EUI11 | EUI10 | EUI09 | EUI08 |

Output signal

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| F084 | EUO07 | EUO06 | EUO05 | EUO04 | EUO03 | EUO02 | EUO01 | EUO00 |
| F085 | EUO15 | EUO14 | EUO13 | EUO12 | EUO11 | EUO10 | EUO09 | EUO08 |

## 6.24 REFERENCING COMMON VARIABLES OF CUSTOM MACROS

An execution, conversational, or auxiliary P-CODE program can reference and write common variables (#100 to #199, #500 to #999) used by a user program for custom macro B. With variables #99100 to #99199 and #99500 to #99999, the P-CODE program can reference and write common variables #100 to #199 and #500 to #999 of the user program.

#99100 → Corresponds to #100.

:

#99149 → Corresponds to #149.

:

#99500 → Corresponds to #500.

:

#99531 → Corresponds to #531.

> **NOTE**
> The common variables #100 to #149 and #500 to #531 can be referenced or written.

## 6.25 DISPLAYING THE CONVERSATIONAL MACRO SCREEN WHEN THE POWER IS TURNED ON

With the conventional compilation parameter, DAUX (No. 9002, #5), the screen of CNC series and edition is displayed while the CNC unit is in the emergency stop state or external reset state. If a new parameter, DAUXR (No. 9006, #3), is specified, the conversational macro screen can be immediately displayed even in the emergency stop state or external reset state.

DAUXR=1: The conversational macro screen is displayed when the power is turned on.
Parameter DAUX (No. 9002, #5) must also be set to 1.

## 6.26 LEAVING THE SCREEN UNCHANGED WHEN THE CUSTOM KEY IS PRESSED

If compilation parameter CNCHG (No. 9006, #4) is set to 1, the execution of a conversational macro can be continued when the CUSTOM key is pressed while the conversational macro screen is displayed. If the parameter is set to 0, the system assumes that a request to change the screen is made and the screen is initialized. The current conversational macro is executed from the beginning.

CNCHG=1: The execution of the conversational macro is continued when the CUSTOM key is pressed while the conversational macro screen is displayed.

## 6.27 FUNCTION FOR FINDING A P–CODE WORK NUMBER

When compilation parameter PWSR (No. 9002, #6) is set to 1, the function for finding a P-CODE work number can be used.

When an operation program created in the program editing memory makes a G-, M-, or T-code call, generally an execution macro (program) is called and executed as a subprogram or a macro program. If automatic operation is started with this function, an execution macro program cataloged in the ROM can be directly activated.

A conversational or auxiliary macro is used to write the number of the P-CODE program to be started in the P-CODE work number control variable (#8610).

If the value of #8610 is not null or 0 when the automatic operation is started in the automatic operation (MEM) mode while the CNC unit is in the reset state, the P-CODE program having the number indicated by the value of #8610 is found and executed from the beginning.

If the value of #8610 is null or zero, the user program currently selected by the CNC unit is executed as usual.

If the program started by this function ends with M99, the currently selected program is executed after M99.

When power is turned on, the value of #8610 is set to 0.

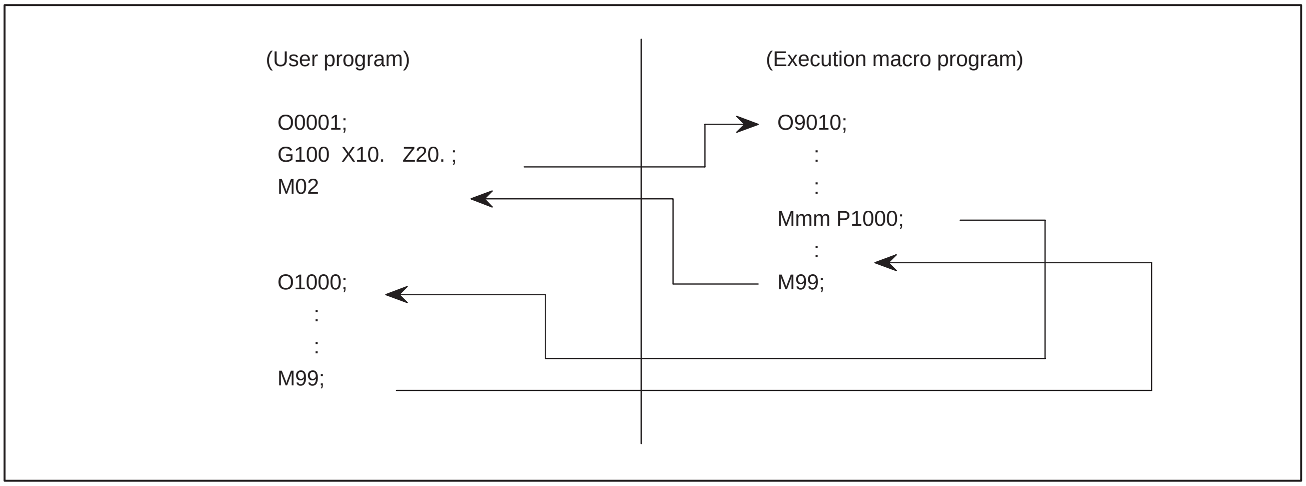
**NOTE**

When #8610 is a value other than zero, this function starts a P-CODE program regardless of the program currently selected by the CNC unit (the user program displayed on the CNC screen). When this function is used, measures to prevent an operator error must be taken. For example, an alarm lamp activated by an auxiliary macro or the PMC and interlock processing are helpful.

## 6.28 FUNCTION FOR CALLING A USER PROGRAM BY AN EXECUTION MACRO

### 6.28.1 Function

Using the M code specified by compilation parameter, a P-CODE program (execution macro program) called by a G, M, or T code from a user program can call another user program as a subprogram.




### 6.28.2 Calling Format

Mmm Ppppp;

mm : M code specified in compilation parameter No. 9033

pppp : Number of the user program to be called
(A variable or expression can be specified.)

### 6.28.3 Multiple Calls

(1) Program call from a user program which was called by an execution macro

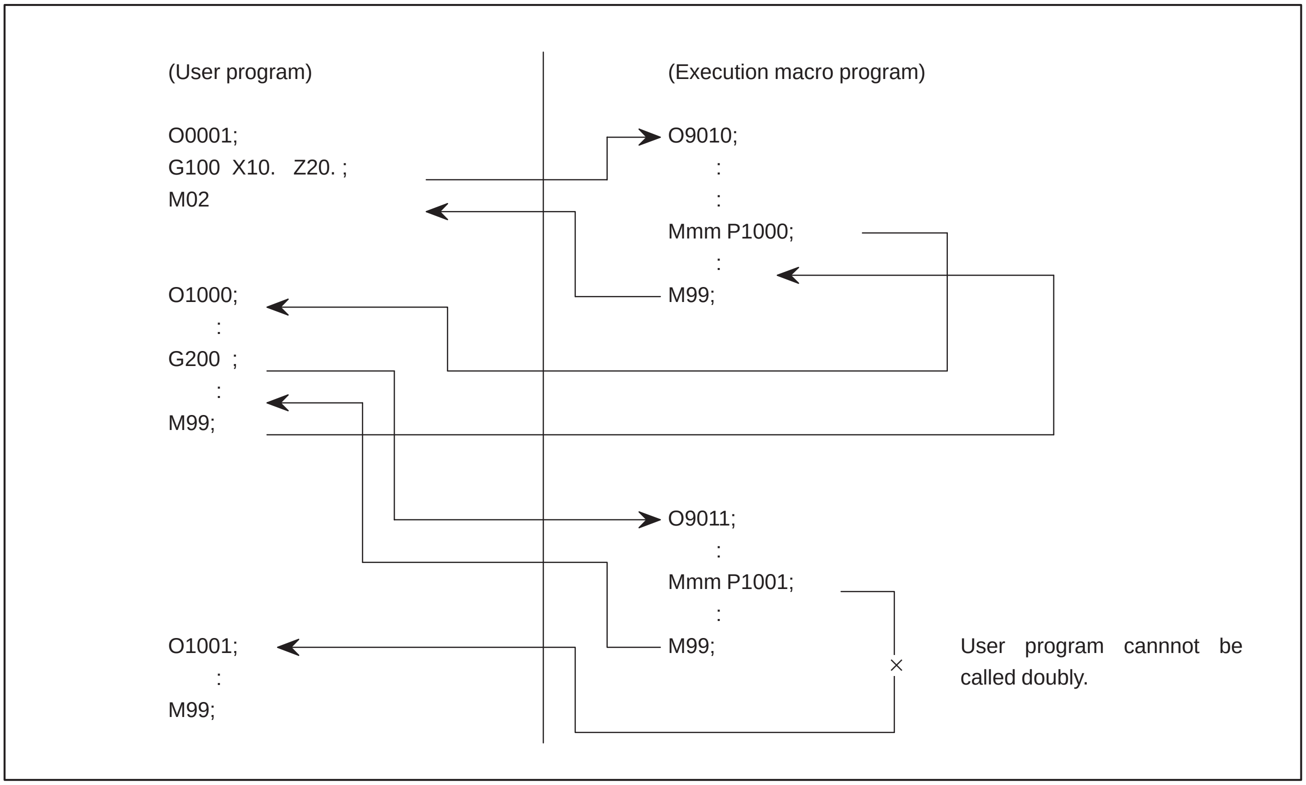
Other user program in the program editing memory can be called from a user program, which was called by an execution macro program, using M98, G65 and G66. However, other user program cannot be called by using G, M, T, or special code, etc.

Also an execution macro program can be called from a user program which was called by an execution macro.

| Program to be called | Calling method | |
|---|---|---|
| | M98, G65, G66 | M/S/T/Special code/etc |
| User program in program editing memory | Yes | No |
| Execution macro program | No | Yes |

(2) Calling from an execution macro

A user program can be called from an execution macro which was called by a user program. However, user program cannot be called doubly from an execution macro program.

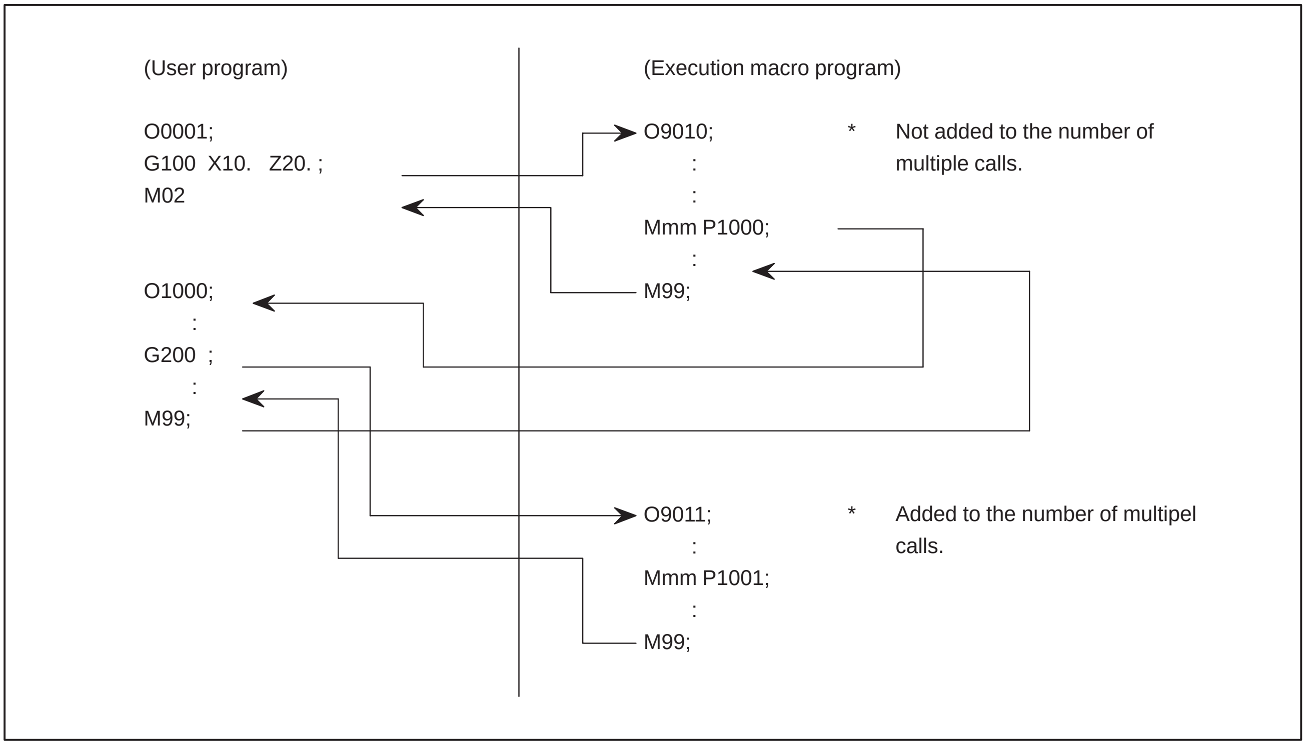



(3) Multiple calls

- The number of multiple calls of user programs and execution macro programs are limited, respectively. The number of multiple calls of user programs called from execution macro programs is calculated independently of the number of multiple calls of execution macros.

| Calling method | Program to be called | |
|---|---|---|
| | User program | Execution macro program |
| Subprogram call | 4 holds | 4 holds |
| Macro call | 4 holds | 4 holds |

- When an execution macro calls a user program, the number of multiple calls of user program is added by one.
- When a user program calls an execution macro program, the number of multiple calls is not added for the first call but it is added by one from the 2nd call.




| Direction of calling | | Addition to the number of multiple calls |
|---|---|---|
| Calling user program from execution macro program | 1st | Not added |
| | 2nd | Added to the number of multiple calls of execution macro program |
| Calling user program from execution macro program | | Added to the number of multiple calls of user program |

**NOTE**
When the program execution returns from a user program to an execution macro program, it is not available to specify a sequence number as a return designation.

# 6.29 OPERATION FUNCTIONS (LOGARITHM, EXPONENT, ARCSINE, ARCCOSINE)

## 6.29.1 Overview

The following operation functions can be used:

| | | |
|---|---|---|
| Logarithm | (LN) | |
| Exponent | (EXP) | |
| Arcsine | (ASIN) | (Unit: Degrees) |
| Arccosine | (ACOS) | (Unit: Degrees) |

The functions above have the same effect as the operation instructions of custom macro B. For the precision, data format, operation precision, and so forth of the functions, refer to the OPERATOR'S MANUAL of each Series.

The functions can be used with an conversational macro, auxiliary macro, or execution macro. When the functions are used with an execution macro, the option for custom macro B is required for the CNC.

## 6.29.2 Operation

To find the value assumed by a function for a given value, enter the given value in #101, then execute the corresponding instruction indicated below. Then, the result is substituted into #101.

Logarithmic function (LN)

```
#100 = LN[#101] ;
```

Exponential function (EXP)

```
#100 = EXP[#101] ;
```

Arcsine function (ASIN)

```
#100 = ASIN[#101] ;
```

Arccosine function (ACOS)

```
#100 = ACOS[#101] ;
```

# 6.30 FUNCTION FOR IDENTIFYING THE PRESSED MDI KEY USING A CONVERSATIONAL MACRO

## 6.30.1 Overview

A control variable has been added so that the MDI number corresponding to the MDI key being pressed can be read using a conversational macro. The control variable allows a conversational macro to identify the MDI key being pressed.

## 6.30.2 Function

By reading the value of control variable #8549, the MDI key being pressed can be identified. This variable holds an MDI number in decimal.

A key number is represented in binary, and is 8 bits long.

This function can also tell the operator whether the key has been pressed together with the SHIFT key or just alone. For detailed information, see section 6.30.3 below.

## 6.30.3 Key Number List

A key number is expressed in binary from 00 to FF.

For example, when the SPACE key is being pressed, the key number 20h is sent to control; variable #8549, which holds 32 in decimal.

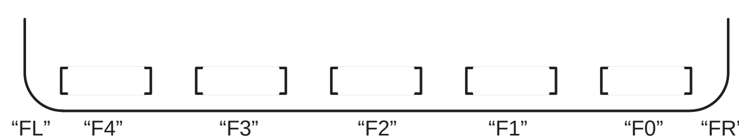
For example, when the RESET key is being pressed, the key number 90h is sent to control variable #8549, which holds 144 in decimal.

**NOTE**

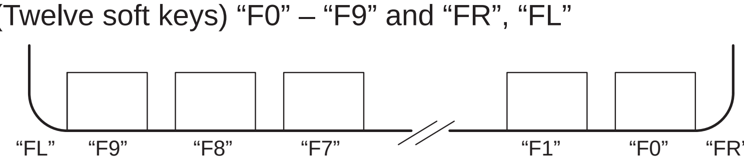
Handling of soft keys

In the key number table, numbers F0 to F9, FR, and FL correspond to soft keys. Numbers other than those above correspond to MDI keys as follows

(Seven soft keys) “F0” – “F4” and “FR”, “FL”




(Twelve soft keys) “F0” – “F9” and “FR”, “FL”



**(00H – 7FH)**

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | | | Space | 0 | @ | P | | |
| 1 | | | | 1 | A | Q | | |
| 2 | | | | 2 | B | R | | |
| 3 | | | # | 3 | C | S | | |
| 4 | | | | 4 | D | T | | |
| 5 | | | | 5 | E | U | | |
| 6 | | | & | 6 | F | V | | |
| 7 | | | | 7 | G | W | | |
| 8 | | | ( | 8 | H | X | | |
| 9 | | | ) | 9 | I | Y | | |
| A | ;<br>(EOB) | | * | | J | Z | | |
| B | | | + | | K | [ | | |
| C | | | ' | | L | | | |
| D | | | – | = | M | ] | | |
| E | | | . | | N | | | |
| F | | | / | ? | O | | | |

**(80H – FFH)**

| | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|
| 0 | | Reset | | | | | | F0 |
| 1 | | | | | | | | F1 |
| 2 | | | | | | | | F2 |
| 3 | | | | | | | | F3 |
| 4 | Shift | Insert | | | | | | F4 |
| 5 | | Delete | | | | | | F5 |
| 6 | CAN | Alter | | | | | | F6 |
| 7 | | | | | | | | F7 |
| 8 | Cur→ | Input | | | | | | F8 |
| 9 | Cur← | | | | | | | F9 |
| A | Cur↓ | Help | | | | | | |
| B | Cur↑ | | | | | | | |
| C | | | | | | | | |
| D | | | | | | | | |
| E | Page↓ | | | | | | | FR |
| F | Page↑ | | | | | | | FL |

# 6.31 WINDOW FUNCTION

## 6.31.1 Overview

The window function can now reference system information.

Conversational macro variable

No.8998: System information ID number

No.8999: System information

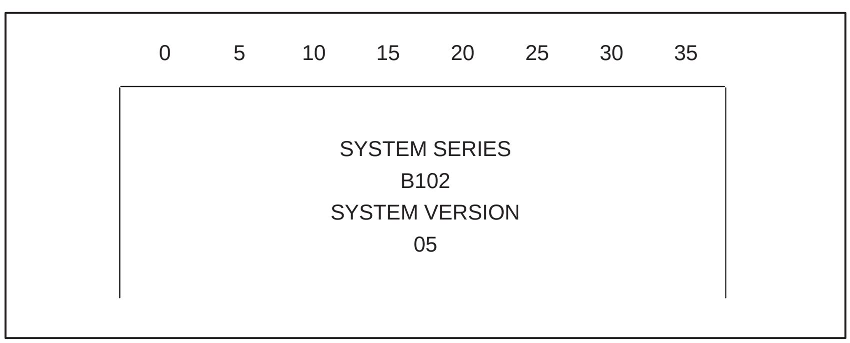
Method of using the function

By setting a desired value in the system information ID number variable (#8998), system information can be read into the system information variable (# 8999).

(Example) Display of system version and editions

```
O6000 ;
G243 X10 Y02 ('SYSTEM SERIES') ;
    #8998 = 8000 ;          /* Get the contents of ID No.8000
G243 X15 Y04 C#8999 ;  /* Display the contens of  ID No.8000
    #8998 = 8001 ;          /* Get the contents of  ID No.8001
G243 X16 Y04 C#8999 ;  /* Display the contents of  ID No.8001
    #8998 = 8002 ;          /* Get the contents of  ID No.8002
G243 X17 Y04 C#8999 ;  /* Display the contents of  ID No.8002
    #8998 = 8003 ;          /* Get the contents of  ID No.8003
G243 X18 Y04 C#8999 ;  /* Display the contents of  ID No.8003
G243 X10 Y06 ('SYSTEM VERSION') ;
    #8998 = 8005 ;          /* Get the contents of  ID No.8005
G243 X15 Y08 C#8999 ;  /* Display the contents of  ID No.8005
    #8998 = 8006 ;          /* Get the contents of  ID No.8006
G243 X16 Y08 C#8999 ;  /* Display the contents of  ID No.8006
```

Above screen is displayed.




> **NOTE**
> System version and edition to be displayed differs depending on NC model.

## 6.31.2
## Referenced System Information

**Referenced System Information and ID No. List (1/4)**

| ID No. | Information | Contents of Information | |
|---|---|---|---|
| 1 | Alarm Information | Alarm check flag | |
| 5 | | Overheat alarm | |
| 6 | | Spindle alarm | No.750 to 763(767) |
| 11 | | P/S Alarm number | |
| 12 | | P/S Alarm number ($\geq$5000) | |
| 13 | | P/S Alarm number (User alarm) | |
| 20 | | Overtravel alarm | No.500 (1st to 8th axis) |
| 21 | | Overtravel alarm | No.501 (1st to 8th axis) |
| 22 | | Overtravel alarm | No.502 (1st to 8th axis) |
| 23 | | Overtravel alarm | No.503 (1st to 8th axis) |
| 24 | | Overtravel alarm | No.504 (1st to 8th axis) |
| 25 | | Overtravel alarm | No.505 (1st to 8th axis) |
| 26 | | Overtravel alarm | No.506 (1st to 8th axis) |
| 27 | | Overtravel alarm | No.507 (1st to 8th axis) |
| 30 | | Servo alarm | No.400 to 407 |
| 31 | | Servo alarm | Axis of servo alarm No.400 (1st to 8th axis) |
| 32 | | Servo alarm | Axis of servo alarm No.401 (1st to 8th axis) |
| 33 | | Servo alarm | Axis of servo alarm No.404 (1st to 8th axis) |
| 35 | | Servo alarm | Judgment of the axis of servo alarm No.401 |
| 36 | | Servo alarm | Axis of servo alarm No.401 (1st to 8th axis) |
| 41 | | Servo alarm 1st axis | No.410 to 417 |
| 42 | | Servo alarm 2nd axis | No.410 to 417 |
| 43 | | Servo alarm 3rd axis | No.410 to 417 |
| 44 | | Servo alarm 4th axis | No.410 to 417 |
| 45 | | Servo alarm 5th axis | No.410 to 417 |
| 46 | | Servo alarm 6th axis | No.410 to 417 |
| 47 | | Servo alarm 7th axis | No.410 to 417 |
| 48 | | Servo alarm 8th axis | No.410 to 417 |
| 55 | | Judgement of external alarm | |
| 56 | | External alarm 1 | |
| 57 | | External alarm 2 | |
| 58 | | External alarm 3 | |
| 59 | | External alarm 4 | |
| 70 | | Judgement of APC alarm display | |
| 71 | | 1st axis of APC alarm | No.300 |
| 72 | | 2nd axis of APC alarm | No.300 |
| 73 | | 3rd axis of APC alarm | No.300 |
| 74 | | 4th axis of APC alarm | No.300 |
| 75 | | 5th axis of APC alarm | No.300 |
| 76 | | 6th axis of APC alarm | No.300 |
| 77 | | 7th axis of APC alarm | No.300 |
| 78 | | 8th axis of APC alarm | No.300 |

**Referenced Systen Information and ID No. List (2/4)**

| ID No. | Information | Contents of Information |
|---|---|---|
| 81 | Alarm Information | 1st axis of APC alarm No.301 to 308 |
| 82 | | 2nd axis of APC alarm No.301 to 308 |
| 83 | | 3rd axis of APC alarm No.301 to 308 |
| 84 | | 4th axis of APC alarm No.301 to 308 |
| 85 | | 5th axis of APC alarm No.301 to 308 |
| 86 | | 6th axis of APC alarm No.301 to 308 |
| 87 | | 7th axis of APC alarm No.301 to 308 |
| 88 | | 8th axis of APC alarm No.301 to 308 |
| 90 | | Judgement of SPC alarm No. 350 to 351 |
| 91 | | SPC alarm No.350 to 351 of 1st axis |
| 92 | | SPC alarm No.350 to 351 of 2nd axis |
| 93 | | SPC alarm No.350 to 351 of 3rd axis |
| 94 | | SPC alarm No.350 to 351 of 4th axis |
| 95 | | SPC alarm No.350 to 351 of 5th axis |
| 96 | | SPC alarm No.350 to 351 of 6th axis |
| 97 | | SPC alarm No.350 to 351 of 7th axis |
| 98 | | SPC alarm No.350 to 351 of 8th axis |
| 100 | Axis, Coordinate and Positional information | No. of CNC controlled axes. |
| 101 | | No. of PMC controlled axes |
| 102 | | Sum of controlled axes |
| 110 | | 1st axis relative coordinate value |
| 111 | | 2nd axis relative coordinate value |
| 112 | | 3rd axis relative coordinate value |
| 113 | | 4th axis relative coordinate value |
| 114 | | 5th axis relative coordinate value |
| 115 | | 6th axis relative coordinate value |
| 116 | | 7th axis relative coordinate value |
| 117 | | 8th axis relative coordinate value |
| 200 | No. of machined parts, etc. | Total no. of parts machinied |
| 201 | | No. of parts requied |
| 202 | | No. of parts machined |
| 210 | | Power on time |
| 220 | | Operation time (Hour, minute) |
| 221 | | Operation time (second) |
| 222 | | Cutting time (Hour, minute) |
| 223 | | Cutting time (Second) |
| 224 | | Timer (Hour, minute) |
| 225 | | Timer (Second) |
| 226 | | Cycle time (Hour, minute) |
| 227 | | Cycle time (Second) |

**Referenced Systen Information and ID No. List (3/4)**

| ID No. | Information | Contents of Information |
|---|---|---|
| 411 | Servo motor load current value | Ist axis servo motor load current value |
| 412 | | 2nd axis servo motor load current value |
| 413 | | 3rd axis servo motor load current value |
| 414 | | 4th axis servo motor load current value |
| 415 | | 5th axis servo motor load current value |
| 416 | | 6th axis servo motor load current value |
| 417 | | 7th axis servo motor load current value |
| 418 | | 8th axis servo motor load current value |
| 700 | Diagnosis Information | Diagnosis 000 to 006 |
| 701 | | Diagnosis 010 to 015 |
| 710 | | Diagnosis 030 |
| 711 | | Diagnosis 031 |
| 712 | | Diagnosis 020 to 025 |
| 800 | | Ist axis position error amount |
| 801 | | 2nd axis position error amount |
| 802 | | 3rd axis position error amount |
| 803 | | 4th axis position error amount |
| 804 | | 5th axis position error amount |
| 805 | | 6th axis position error amount |
| 806 | | 7th axis position error amount |
| 807 | | 8th axis position error amount |
| 8000 | System configuration | System version 4th digit Main |
| 8001 | | System version 3rd digit Main |
| 8002 | | System version 2nd digit Main |
| 8003 | | System version 1st digit Main |
| 8005 | | System edition 2nd digit Main |
| 8006 | | System edition 1st digit Main |
| 8010 | | System version 4th digit Sub (TT only) |
| 8011 | | System version 3rd digit Sub (TT only) |
| 8012 | | System version 2nd digit Sub (TT only) |
| 8013 | | System version 1st digit Sub (TT only) |
| 8015 | | System edition 2nd digit Sub (TT only) |
| 8016 | | System edition 1st dgt Sub (TT only) |
| 8020 | | Servo series 4th digit |
| 8021 | | Servo series 3rd digit |
| 8022 | | Servo series 2nd digit |
| 8023 | | Servo series 1st digit |
| 8025 | | Servo edition 4th digit |
| 8026 | | Servo edition 3rd digit |
| 8030 | | PMC version 4th digit |
| 8031 | | PMC version 3rd digit |
| 8032 | | PMC version 2nd digit |
| 8033 | | PMC version 1st digit |

**Referenced Systen Information and ID No. List (4/4)**

| ID No. | Information | Contents of Information |
|---|---|---|
| 8030 | System configuration | PMC version 4th digit |
| 8031 | | PMC version 3rd digit |
| 8032 | | PMC version 2nd digit |
| 8033 | | PMC version 1st digit |
| 8035 | | PMC edition 2nd digit |
| 8036 | | PMC edition 3rd digit |
| 8040 | | Ladder version 4th digit |
| 8041 | | Ladder version 3rd digit |
| 8042 | | Ladder version 2nd digit |
| 8043 | | Ladder version 1st digit |
| 8045 | | Ladder edition 2nd digit |
| 8046 | | Ladder edition 1st digit |
| 8050 | | Order made macro version 4th digit |
| 8051 | | Order made macro version 3rd digit |
| 8052 | | Order made macro version 2nd digit |
| 8053 | | Order made macro version 1st digit |
| 8055 | | Order made macro edition 2nd digit |
| 8056 | | Order made macro edition 1st digit |
| 8200 | | Main board CRTC information |
| 8201 | | Optional board CRTC information |
| 8202 | | Graphic module 1 information |
| 8203 | | Graphic module 2 information |

## 6.31.3 Detailed Description of Reference System Information

Alarm information

Axis/coordinate position information

Miscellaneous information including the total number of machined parts

Diagnosis information

System configuration information

(1) Alarm information

| | | |
|---|---|---|
| (a) | Alarm check flag | ID No. 1 |
| (b) | Overheat alarms | ID No. 5 |
| (c) | Spindle alarms | ID No. 6 |
| (d) | P/S alarms | ID Nos. 11 to 13 |
| (e) | Overtravel alarms | ID Nos. 20 to 27 |
| (f) | Servo alarms | ID Nos. 30 to 48 |
| (g) | External alarms | ID Nos. 55 to 59 |
| (h) | Absolute pulse coder (APC) alarms | ID Nos. 70 to 98 |

First check the information of ID No. 1, then proceed to the processing of each alarm.

(Example)

```
O6001 ;
        #501  = 1 ;
        #502  = 1 ;
        #8998 = 1 ;
N10     #500  = #8999 ;            /* Contents of ID No.1 is taken.
        #500  = #500 AND #501 ;
IF[#500 NE    0] GOTO [#502*100] ;       /* Alarm ?
IF[#501 GE 32768] GOTO 900 ;
        #501 = #501*2 ;
        #502 = #502+1 ;
                        GOTO 10  ;
N100 (Processing Bit 0001h )
          :         :
N200 (Processing Bit 0002h )
          :         :
N300 (Processing Bit 0004h )
          :         :
          :         :
          :         :
          :         :
          :         :
N900 M99 ;
```

(a) Alarm check flag

The alarm check flag ID No. 1, is a one-word data representing a number from 0 to 65535.

This flag indicates which alarm was issued.

Example : When bit 3 is 1 (on), 8 is output.
When bit 9 is 1 (on), 512 is output.
When bit 4 and bit 9 are 1 (on), 528 is output.

| | #f | #e | #d | #c | #b | #a | #9 | #8 | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 | Output information |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit (0001h) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| Bit (0002h) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 2 |
| Bit (0004h) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 4 |
| Bit (0008h) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 8 |
| Bit (0010h) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 16 |
| Bit (0020h) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 32 |
| Bit (0050h) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 64 |
| Bit (0080h) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 128 |
| Bit (0100h) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 256 |
| Bit (0200h) | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 512 |
| Bit (0400h) | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1024 |
| Bit (0800h) | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2048 |
| Bit (1000h) | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4096 |
| Bit (2000h) | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 80192 |
| Bit (4000h) | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 16384 |
| Bit (8000h) | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 32768 |

For the meaning of each bit, see the alarm check flag table.

**Alarm Check Flag**

| ID No. | Bit Information | Meaning |
|---|---|---|
| 1 | Bit (0001h) | PS alarm No.100. Parameter is write enable. |
| | Bit (0002h) | P/S alarm No.000. Turn on the power again. |
| | Bit (0004h) | P/S alarm No.101. Power was turned off during program editing. Press <PROG> key and <RE-SET> key. |
| | Bit (0008h) | P/S alarm |
| | Bit (0010h) | Overtravel alarm |
| | Bit (0020h) | Overheat alarm |
| | Bit (0040h) | Servo alarm |
| | Bit (0080h) | Memory alarm |
| | Bit (0100h) | APC alarm |
| | Bit (0200h) | Spindle alarm |
| | Bit (0400h) | P/S alarm (No.500 or more) |
| | Bit (0800h) | F-16 laser alarm |
| | Bit (1000h) | ———————— |
| | Bit (2000h) | ———————— |
| | Bit (4000h) | ———————— |
| | Bit (8000h) | External alarm |

(b) Overheat alarms

A one-byte data, ID No. 5, representing a number from 0 to 255 indicates which overheat alarm was issued.

Example : When bit 3 is 1 (on), 8 is output.
When bit 5 is 1 (on), 32 is output.
When bit 3 and bit 5 are 1 (on), 40 is output.

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 | Output information |
|---|---|---|---|---|---|---|---|---|---|
| Bit (01h) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| Bit (02h) | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 2 |
| Bit (04h) | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 4 |
| Bit (08h) | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 8 |
| Bit (10h) | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 16 |
| Bit (20h) | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 32 |
| Bit (40h) | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 64 |
| Bit (80h) | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 128 |

For the meaning of each bit, see the overheat alarms table.

**Overheat Alarm Table**

| ID No. | Bit information | Alarm No. | Meaning |
|---|---|---|---|
| 5 | Bit (01h) | 700 | Overheat of master PCB |
| | Bit (02h) | ———— | ———————— |
| | Bit (04h) | ———— | ———————— |
| | Bit (08h) | 704 | Spindle overheat by spindle speed fluctuation detection |
| | Bit (10h) | 701 | FAN MOTER |
| | Bit (20h) | ———— | ———————— |
| | Bit (40h) | ———— | ———————— |
| | Bit (80h) | ———— | ———————— |

(c) Spindle alarms

A one-word data, ID No. 6, representing a number from 0 to 65535 indicates which spindle alarm was issued.

For one-word data, see Section 6.31.3.(1)(a).

For the meaning of each bit, see the spindle alarm table.

**Spindle Alarm Table**

| ID No. | Bit information | Alarm No. | Meaning |
|---|---|---|---|
| 6 | Bit (0001h) | 749 | S-SPINDLE LSI ERROR |
| | Bit (0002h) | 750 | This alarm is issued if the spindle control unit of a system with serial spindles is not started normally when the power is turned on. Four causes can be considered:<br>1) The optical cable makes poor contact, or power to the main control unit is turned off.<br>2) Power to the NC is turned on when the LED display of the spindle control unit indicates SU-01 or when an alarm state other than AL-24 is present.<br>3) Other causes (such as an incorrect hardware combination)<br>4) The second spindle (when bit 4 (SP2) of parameter No. 3701 is set to 1) is in one of the states 1) to 3) above. |
| | Bit (0004h) | 751 | This alarm is output to the NC to indicate alarm generation on the spindle unit of a system with serial spindles. Alarm information is indicated by AL-xx (xx = number). For detailed alarm information, see the AC Spindle Servo Unit Maintenance Manual. |
| | Bit (0008h) | 752 | In serial spindle control, switching to the contouring mode, spindle positioning mode, rigid tapping mode, or spindle control mode is not terminated normally. (This alarm is issued when the spindle control unit reacts abnormally to a switching command from the NC). |
| | Bit (0010h) | 753 | SPD-1 DATA TRNSFER ERROR |
| | Bit (0020h) | ———— | ———— |
| | Bit (0040h) | ———— | ———— |
| | Bit (0080h) | ———— | ———— |
| | Bit (0100h) | ———— | ———— |
| | Bit (0200h) | 761 | Refer to Alarm No.751 |
| | Bit (0400h) | 762 | Refer to alarm No.752 |
| | Bit (0800h) | 763 | SPD-2 DATA TRNSFER ERROR |
| | Bit (1000h) | ———— | ———— |
| | Bit (2000h) | ———— | ———— |
| | Bit (4000h) | ———— | ———— |
| | Bit (8000h) | ———— | ———— |

(d) Program/setting (P/S) alarms

The information of ID Nos. 11 and 12 directly indicates the P/S alarm number.

The information of ID No. 13 is valid only when ID No. 11 indicates 255. The information of ID No. 13 is added to 2500 to make a P/S alarm number.

**P/S Alarm Table**

| ID No. | Meaning |
|---|---|
| 11 | P/S alarm No. |
| 12 | P/S alarm No. (5000 or more) |
| 13 | P/S alarm No. (user alarm) |

(e) Overtravel alarms

Each of ID Nos. 20 to 27 corresponds to an alarm number. The Information of each ID number indicates the axis in which the alarm occurred.

For one-byte data, see Section 6.31.3.(1)(b).

For the meaning of each bit, see the overtravel alarm table.

**Overtravel Alarm Table**

| ID No. | Alarm No. | Meaning |
|---|---|---|
| 20 | 500 | Stored stroke limit I of + side was passed. |
| 21 | 501 | Stored stroke limit I of – side was passed. |
| 22 | 502 | Stored stroke limit II of + side was passed. |
| 23 | 503 | Stored stroke limit II of – side was passed. |
| 24 | 504 | Stored stroke limit III of + side was passed. |
| 25 | 505 | Stored stroke limit III of – side was passed. |
| 26 | 506 | Hardware OT of + side was passed. |
| 27 | 507 | Hardware OT of – side was passed. |

**Overtravel Alarm Table**

| ID No. | Bit information | Meaning |
|---|---|---|
| 20 - 27 | Bit (01h) | Overtravel alrm of 1st axis |
| | Bit (02h) | Overtravel alarm of 2nd axis |
| | Bit (04h) | Overtravel alarm of 3rd axis |
| | Bit (08h) | Overtravel alarm of 4th axis |
| | Bit (10h) | Overtravel alarm of 5th axis |
| | Bit (20h) | Overtravel alarm of 6th axis |
| | Bit (40h) | Overtravel alarm of 7th axis |
| | Bit (80h) | Overtravel alarm of 8th axis |

(f) Servo alarms

The information of ID No. 30 indicates servo alarm Nos. 400 to 407. ID No. 31 indicates the number of the axis in which alarm No. 400 occurred, ID No. 32 indicates the number of the axis in which alarm No. 401 occurred, and ID No. 33 indicates the number of the axis in which alarm No. 404 occurred.

Note, however, that the number of the axis in which alarm No. 401 occurred serves as the information of ID No. 36 when the logical product of the information of ID No. 32 and the information of ID No. 35 is 0.

Each of ID Nos. 41 to 48 corresponds to a servo alarm axis, and the information of each ID number indicates an alarm number.

All information consists of one-byte data. For one-byte data, see Section 6.31.3.(1)(b). For the bit configuration, see the servo alarm table.

**Servo Alarm Table**

| ID No. | Bit information | Alarm No. | Meaning |
|---|---|---|---|
| 30 | Bit (01h) | 400 | Overload signal is turned on. |
| | Bit (02h) | 401 | Ready signal (DRDY) of servo amplifier turned off. |
| | Bit (04h) | ——— | ———— |
| | Bit (08h) | ——— | ———— |
| | Bit (10h) | 404 | The ready signal (MCON) is off but the ready signal (DRDY) of the servo amplifier is still on. Alternatively, when power is turned on, DRDY is on but MCON is still off. |
| | Bit (20h) | 405 | The position control system is abnormal. Reference position return may have failed due to an abnormality in the NC or servo system. Retry, starting with reference position return. |
| | Bit (40h) | ——— | ———— |
| | Bit (80h) | 407 | Difference of position error between synchronized axes exceeds the specified value. |

**Servo Alarm Table**

| ID No. | Bit Information | Meaning |
|---|---|---|
| 31 – 33, 36 | Bit (01h) | 1st axis servo alarm |
| | Bit (02h) | 2nd axis servo alarm |
| | Bit (04h) | 3rd axis servo alarm |
| | Bit (08h) | 4th axis servo alarm |
| | Bit (10h) | 5th axis servo alarm |
| | Bit (20h) | 6th axis servo alarm |
| | Bit (40h) | 7th axis servo alarm |
| | Bit (80h) | 8th axis servo alarm |

| ID No. | Meaning |
|---|---|
| 35 | Judgement to fix an of servo alarm No.401 |

| ID No. | Meaning |
|---|---|
| 41 | Servo alarm 1st axis |
| 42 | Servo alarm 2nd axis |
| 43 | Servo alarm 3rd axis |
| 44 | Servo alarm 4th axis |
| 45 | Servo alarm 5th axis |
| 46 | Servo alarm 6th axis |
| 47 | Servo alarm 7th axis |
| 48 | Servo alarm 8th axis |

**Servo Alarm Table**

| ID No. | Bit information | Alarm No. | Meaning |
|---|---|---|---|
| 41 - 48 | Bit (01h) | 411 | Position error during movement is larger than set value. |
| | Bit (02h) | 413 | Error register contents exceeds $\pm 2{**}31$. |
| | Bit (04h) | 415 | A speed larger than 511875 detection units/sec is to be specified. |
| | Bit (08h) | 416 | Position detection system of pulse coder is abnormal. |
| | Bit (10h) | 412 | ———————— |
| | Bit (20h) | 410 | Position error at stop is larger than set value. |
| | Bit (40h) | 414 | Digital servo system is abnormal. |
| | Bit (80h) | 417 | This alarm is issued if any of the following conditions occur:<br>1) A value beyond the specifiable range is set in parameter No. 2020 (motor type).<br>2) A correct value (111 or -111) is not set in parameter No. 2022 (motor rotation direction).<br>3) An incorrect value such as a negative value is set in parameter No. 2023 (number of speed feedback pulses per motor revolution).<br>4) An incorrect value such as a negative value is set in parameter No. 2024 (number of position feedback pulses per motor revolution).<br>5) Parameter Nos. 2084 and 2085 (flexible feed gear ratios) are not set.<br>6) A value beyond a range from 1 to the number of controlled axes, or a value not successive is set in parameter No. 1023 (servo axis number). |

(g) External alarms

A one-byte data, ID No. 55, representing a number from 0 to 255 indicates which external alarm was issued.

The information of ID Nos. 56 to 59 directly indicates alarm numbers.

For one-byte data, see Section 6.31.3.(1)(b). For the meaning of each bit, see the external alarm table.

**External Alarm Table**

| ID No. | Bit information | Meaning |
|---|---|---|
| 55 | Bit (01h) | External alarm 1 |
| | Bit (02h) | External alarm 2 |
| | Bit (04h) | External alarm 3 |
| | Bit (08h) | External alrm 4 |
| | Bit (10h) | ———————— |
| | Bit (20h) | ———————— |
| | Bit (40h) | ———————— |
| | Bit (80h) | ———————— |

| ID No. | Meaning |
|---|---|
| 56 | Alarm No. of external alarm 1 |
| 57 | Alarm No. of external alarm 2 |
| 58 | Alarm No. of external alarm 3 |
| 59 | Alarm No. of external alarm 4 |

(h) Absolute pulse coder (APC) alarms

First check ID No. 70. If bit 0 is on, check the following:

ID No. 71 to 78

ID No. 81 to 88

ID No. 90 to 98

Each of ID No. 71 to 78 corresponds to an axis; bit 5, if on, indicates that alarm No. 300 has occurred.

Each of ID No. 81 to 88 corresponds to an axis; the information of each ID No. indicates an APC alarm number.

Check ID No. 90 to determine whether to read the information of ID No. 91 to 98 indicating serial pulse coder (SPC) alarm information. Correct information can be read if bit 2 is on. Each of ID No. 91 to 98 corresponds to an axis.

All information consists of one-byte data. For one-byte data, see Section 6.31.3.(1)(b). For the meaning of each bit, see the APC alarms table.

**APC Alarm Table**

| ID No. | Bit information | Meaning |
|---|---|---|
| 70 | Bit (01h) | Displays APC alarm |

| ID No. | Meaning |
|---|---|
| 71 | 1st axis alarm No.300 |
| 72 | 2nd axis alarm No.300 |
| 73 | 3rd axis alarm No.300 |
| 74 | 4th axis alarm No.300 |
| 75 | 5th axis alarm No.300 |
| 76 | 6th axis alarm No.300 |
| 77 | 7th axis alarm No.300 |
| 78 | 8th axis alarm No.300 |

**APC Alarm Table**

| ID No. | Bit information | Meaning |
|---|---|---|
| 71 – 78 | Bit (01h) | ———————— |
| | : | : : |
| | Bit (10h) | ———————— |
| | Bit (20h) | Manual reference position return is required. |
| | Bit (40h) | ———————— |
| | Bit (80h) | ———————— |

| ID No. | Meaning |
|---|---|
| 81 | 1st axis APC alarm |
| 82 | 2nd axis APC alarm |
| 83 | 3rd axis APC alarm |
| 84 | 4th axis APC alarm |
| 85 | 5th axis APC alarm |
| 86 | 6th axis APC alarm |
| 87 | 7th axis APC alarm |
| 88 | 8th axis APC alarm |

| ID No. | Bit information | Alarm No. | Meaning |
|---|---|---|---|
| 81 – 88 | Bit (01h) | 301 | APC communication error (Data transmission fault) |
| | Bit (02h) | 302 | APC over time error (Data transmission fault) |
| | Bit (04h) | 303 | APC framing error (Data transmission fault) |
| | Bit (08h) | 304 | APC Parity error (Data transmission fault) |
| | Bit (10h) | 305 | APC Pulse missing alarm.(APC alarm) |
| | Bit (20h) | 306 | APC battery voltage has lowered below the level, in which no data is held. (APC alarm) |
| | Bit (40h) | 307 | APC battery voltage is a level that requires replacement. (APC alarm) |
| | Bit (80h) | 308 | APC battery voltage might became a level that requires battery replacement in the past (APC alarm). |

**APC Alarm Table**

| ID No. | Bit information | Meaning |
|---|---|---|
| 90 | Bit (04h) | Displays SPC alarm |

| ID No. | Meaning |
|---|---|
| 91 | 1st axis SPC alarm |
| 92 | 2nd axis SPC alarm |
| 93 | 3rd axis SPC alarm |
| 94 | 4th axis SPC alarm |
| 95 | 5th axis SPC alarm |
| 96 | 6th axis SPC alarm |
| 97 | 7th axis SPC alarm |
| 98 | 8th axis SPC alarm |

**APC Alarm Table**

| ID No. | Bit information | Alarm No. | Meaning |
|---|---|---|---|
| 91 – 98 | Bit (01h) | 350 | Abnormal serial pulse coder. |
| | Bit (02h) | 351 | Communication error of serial pulse coder (Data transmission fault) |
| | Bit (04h) | ———— | —————— |
| | Bit (08h) | ———— | —————— |
| | Bit (10h) | ———— | —————— |
| | Bit (20h) | ———— | —————— |
| | Bit (40h) | ———— | —————— |
| | Bit (80h) | ———— | —————— |

(2) Detailed information of Axis and Coordinate

**Axis and Coordinate Detailed Information**

| ID No. | Meaning |
|---|---|
| 100 | No. of CNC controlled axes |
| 101 | No. of PMC controlled axes |
| 102 | No. of axes |
| 110 | 1st axis relative coordinate value |
| 111 | 2nd axis relative coordinate value |
| 112 | 3rd axis relative coordinate value |
| 113 | 4th axis relative coordinate value |
| 114 | 5th axis relative coordinate value |
| 115 | 6th axis relative coordinate value |
| 116 | 7th axis relative coordinate value |
| 117 | 8th axis relative coordinate value |

(3) Miscellaneous information including the total number of machined parts

The information of ID Nos. 210, 220, 222, 224, and 226 is given in minutes.

Example: When ID No. 220 indicates 360, this information means 6 hours.
When ID No. 220 indicates 369, this information means 6 hours and 9 minutes.
When ID No. 224 indicates 359, this information means 5 hours and 59 minutes.

The information of ID Nos. 221, 223, 225, and 227 is given in 1/1000 seconds.

Example: When ID No. 221 indicates 3000, this information means 3 seconds.
When ID No. 221 indicates 36000, this information means 36 seconds.

**No. of Total Parts Machined, etc.**

| ID No. | Meaning |
|---|---|
| 200 | Total number of Parts machined |
| 201 | Number of parts required |
| 202 | Number of parts machined |
| 210 | Power on time |
| 220 | Operation time |
| 221 | Operation time |
| 222 | Cutting time |
| 223 | Cutting time |
| 224 | General purpose integrated time |
| 225 | General purpose integrated time |
| 226 | Cycle time |
| 227 | Cycle time |

(4) Servo motor load current (Series 16/18)

A load current, digitized by the A/D converter, can be read. The read data is input as a value between –7282.0 and +7282.0 inclusive.

| ID No. | Meaning |
|---|---|
| 411 | 1st axis servo motor load current |
| 412 | 2nd axis servo motor load current |
| 413 | 3rd axis servo motor load current |
| 414 | 4th axis servo motor load current |
| 415 | 5th axis servo motor load current |
| 416 | 6th axis servo motor load current |
| 417 | 7th axis servo motor load current |
| 418 | 8th axis servo motor load current |

Details of read data

The actual load current is calculated as follows:

(AD $\times$ N)/7282 = Load current (A peak)

AD: Input value (value of #8999)

N:

<table>
<tr><th>Motor type</th><th>N value</th><th>Motor type</th><th>N value</th></tr>
<tr><td>4-0S<br>3-0S</td><td>4</td><td rowspan="2">5S/3000<br>10S/3000<br>20S/3000<br>30S/1200</td><td rowspan="2">80</td></tr>
<tr><td rowspan="2">1-0S<br>2-0S<br>1-0S/3000</td><td rowspan="2">12</td></tr>
<tr><td>30/2000</td><td>100</td></tr>
<tr><td>0S<br>5S<br>10S<br>20S/1500</td><td>40</td><td>20S/3000<br>30S/3000<br>40S/2000</td><td>130</td></tr>
</table>

The load current calculated from the above formula is the value corresponding to the maximum current for the motor being used. The calculated load current, therefore, becomes smaller than the continuous rated current for the motor, as displayed on the servo adjustment screen.

(5) Diagnosis information

Each of one-byte data, ID Nos. 700, 701, and 712, representing a number from 0 to 255 indicates diagnosis information.

Example: When bit 3 is 1 (on), 8 is output.
When bit 5 is 1 (on), 32 is output.
When bit 3 and bit 5 are 1 (on), 40 is output.

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 | Output information |
|---|---|---|---|---|---|---|---|---|---|
| Bit (01h) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| Bit (02h) | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 2 |
| Bit (04h) | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 4 |
| Bit (08h) | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 8 |
| Bit (10h) | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 16 |
| Bit (20h) | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 32 |
| Bit (40h) | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 64 |
| Bit (80h) | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 128 |

For the meaning of each bit, see the diagnosis information table.

**Diagnose Detailed Information**

| ID No. | Bit Information | Meaning |
|---|---|---|
| 700 | Bit (01h) | M,S,T function is being executed. |
| | Bit (02h) | Move command is being executed in automatic operation. |
| | Bit (04h) | Dwell is being executed. |
| | Bit (08h) | Inposition check is being done. |
| | Bit (10h) | Feed rate override is 0%. |
| | Bit (20h) | Interlock is on. |
| | Bit (40h) | Waiting for spindle speed arrival signal becom on. |
| | Bit (80h) | ———————— |
| 701 | Bit (01h) | Data is being output via reader/puncher interface. |
| | Bit (02h) | Data is being input via reader/puncher interface. |
| | Bit (04h) | Waiting for index table clamping /unclamping befor/after B axis index table indexing. |
| | Bit (08h) | ———————— |
| | Bit (10h) | Jog override is 0 %. |
| | Bit (20h) | Emergency stop, external reset, reset & rewind or reset key on MDI panel is on. |
| | Bit (40h) | External program number is being searched. |
| | Bit (80h) | ———————— |

**Diagnose Detailed Information**

| ID No. | System information | Meaning |
|---|---|---|
| 710 | 0 – 255 | Displays the character that caused TH alarm by the number of characters from the top of the block. |
| 711 | 0 – 255 | Reading code of the character that caused TH alarm. |

**Diagnose Detailed Information**

| ID No. | Bit Information | Meaning |
|---|---|---|
| 712 | Bit (01h) | Emergency stop or servo alarm. |
| | Bit (02h) | ———————— |
| | Bit (04h) | ———————— |
| | Bit (08h) | Reset key is pressed. |
| | Bit (10h) | Reset & rewind turned on. |
| | Bit (20h) | Emergency stop |
| | Bit (40h) | External reset, emergency stop reset, or reset & rewind |
| | Bit (80h) | A flag to stop pulse distribution by any of the following actions:<br>1) External reset<br>2) Reset & rewind<br>3) Emergency stop<br>4) Feed hold<br>5) Reset key on MDI<br>6) Mode changed to a manual mode (JOG/HANDLE/INC)<br>7) An alarm is issued |

| ID No. | Meaning |
|---|---|
| 800 | 1st axis position error |
| 801 | 2nd axis position error |
| 802 | 3rd axis position error |
| 803 | 4th axis position error |
| 804 | 5th axis position error |
| 805 | 6th axis position error |
| 806 | 7th axis position error |
| 807 | 8th axis position error |

(6) System configuration information

The system information of the following ID numbers is output as ASCII code information represented in decimal:

8000 - 8003, 8005 - 8006

8010 - 8013, 8015 - 8016

8020 - 8023, 8025 - 8026

8030 - 8033, 8035 - 8036

8040 - 8043, 8045 - 8046

8050 - 8053, 8055 - 8056

**NOTE**

ID Nos. 8010 to 8013, 8015m and 8016 are usable only with the TT system.

For the meanings of ID Nos. 8200 to 8211, see the system configuration information table.

(Series 16/18)

| Contents | Hexadecimal | Decimal | Contents | Hexadecimal | Decimal | Contents | Hexadecimal | Decimal |
|---|---|---|---|---|---|---|---|---|
| A | 41 | 65 | N | 4e | 78 | 0 | 30 | 48 |
| B | 42 | 66 | O | 4f | 79 | 1 | 31 | 49 |
| C | 43 | 67 | P | 50 | 80 | 2 | 32 | 50 |
| D | 44 | 68 | Q | 51 | 81 | 3 | 33 | 51 |
| E | 45 | 69 | R | 52 | 82 | 4 | 34 | 52 |
| F | 46 | 70 | S | 53 | 83 | 5 | 35 | 53 |
| G | 47 | 71 | T | 54 | 84 | 6 | 36 | 54 |
| H | 48 | 72 | U | 55 | 85 | 7 | 37 | 55 |
| I | 49 | 73 | V | 56 | 86 | 8 | 38 | 56 |
| J | 4a | 74 | W | 57 | 87 | 9 | 39 | 57 |
| K | 4b | 75 | X | 58 | 88 | | | |
| L | 4c | 76 | Y | 59 | 89 | | | |
| M | 4d | 77 | Z | 5a | 90 | | | |

**System Configuration Information Table**

| ID No. | Bit information | Meaning |
|---|---|---|
| 8200 | 0 | 14-inch CRT screen module is installed |
| | 1 | 9-inch CRT screen module is installed |
| | 2 | 10-inch CRT screen module is installed |
| | 255 | CRT screen module is not installed |
| 8201 | 0 | 14-inch CRTC screen main module is installed |
| | 1 | 9-inch CRTC screen main module is installed |
| | 2 | 10-inch CRTC screen main module is installed |
| | 4 | 14-inch CRTC screen graphic module is installed |
| | 5 | 9-inch CRTC screen graphic module is installed |
| | 6 | 10-inch CRTC screen graphic module is installed |
| | 255 | CRT screen module is not installed |
| 8202 | 0 | Graphic module is installed |
| | 255 | Graphic module is not installed |
| 8203 | 0 | Graphic module is installed |
| | 255 | Graphic module is not installed |
| 8210 | 255 | No CRT module is mounted. |
| | 1 | 9” CRT module is mounted. |
| | 2 | 9” color CRT module is mounted. |
| | 3 | 14” color CRT module is mounted. |
| | 4 | LCD 14” color CRT module is mounted. |
| | 5 | LCD 9” CRT module is mounted. |
| | 6 | VGA 14” color CRT module is mounted. |
| | 7 | Undefined |
| | 8 | LCD 14” CRT module is mounted. |
| | 9 | LCD 9” color CRT module is mounted. |
| | 12 | VGA 9” CRT module is mounted. |
| | 10 | VGA 9” color CRT module is mounted. |
| | 11 | VGA 14” CRT module is mounted. |
| 8211 | 255 | No graphics module is mounted. |
| | 1 | standard graphics module is mounted. |
| | 2 | FAPT graphics module is mounted. |
| | 3 | MMC graphics module is mounted. |
| | 4 | Undefined |
| | 5 | VGA graphics module is mounted. |

## 6.32 EXTERNAL CHARACTER REGISTRATION/ DISPLAY FUNCTION (Series 20)

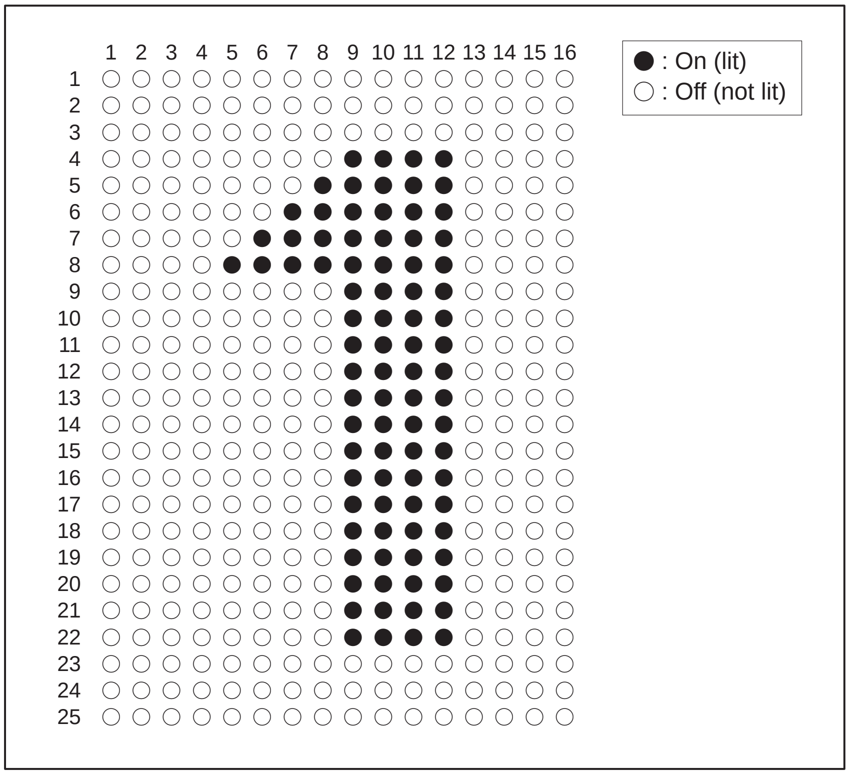
External characters are character patterns generated and registered by the user. With the Series 20, the user can register up to 40 external characters for display using conversational macros. This function allows the user to display special characters and graphics other than the standard character patterns provided by the CNC.

The methods of registering and displaying external characters are described below.

> **NOTE**
> With the Series 16/18/21, the external character registration/display function cannot be used.

(1) Registering external characters

A character on the CRT screen of the Series 20 consists of 400 pixels: 16 pixels (horizontal) by 25 pixels (vertical).




To register an external character, 25 variables are used.

One variable represents the states of 16 horizontal pixels, on or off, in BCD rotation. A row of 25 variables represents one character (400 pixels).

Example:)

○○○○○○○○○○○○○○○○ = 0000000000000000 = 0

○○○○○○○○○○○○●●○○ = 0000000000001100 = 12

●○○○○○○○○○○○○○○○ = 1000000000000000 = 32768

●●●●●●●●●●●●●●●● = 1111111111111111 = 65535

A character pattern defined by a row of 25 variables is stored in external character memory using a G code in the format below.

<Command format>

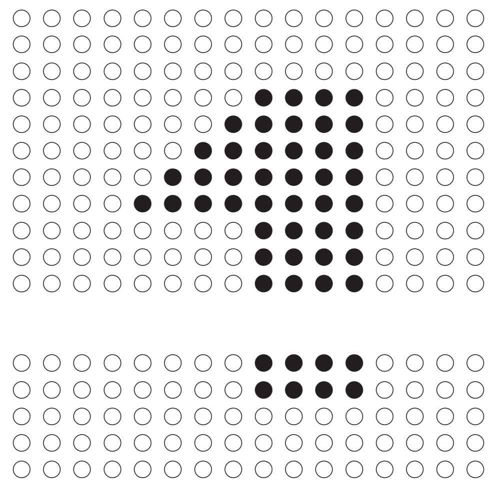
<u>G319 Ppppp Qqq ;</u>

pppp : Number of the first variable of a row of 25 variables defining a character pattern

qq : External character number (00 to 39)
(External memory is as large as 40 characters, and qq specifies a character position for storing a character pattern.))

Example: Macro program for storing the number 1 indicated above as the 20th character in external character memory (external character number = 20)

```
#101= 0 ;
#102= 0 ;
#103= 0 ;
#104= 240 ;
#105= 496 ;
#106= 1008 ;
#107= 2032 ;
#108= 4080 ;
#109= 240 ;
#110= 240 ;
#111= 240 ;
  :  =   :
  :  =   :
#121= 240 ;
#122= 240 ;
#123= 0 ;
#124= 0 ;
#125= 0 ;
```



```
G319 P101 Q20 ;
```

**NOTE**

1 By executing G319, the character pattern defined by a row of 25 variables is stored in external character memory. After a character pattern is stored, the 25 variables can be used for other purposes.
2 The data in external character memory is erased when power is turned off; registered character patterns are erased.
3 Specify an external character number from 00 to 39. If a number not within this range is specified, the registration/display function does not function normally.

(2) Displaying external characters

By executing a G code in the format below, an external character stored in external character memory can be displayed.

<Command format>

G243 Xx Yy (“80qq”) ;

Xx, Yy: Specifies a display start position. X and Y represent an X coordinate and Y coordinate in the character coordinate system, respectively.

(“80qq”): Specifies the external character number of an external character to be displayed. Note, however, that an external character number from 00 to 39 must be specified for qq in hexadecimal. 80qq must be enclosed in quotation marks inside of parentheses.
That is, 80qq must be specified in the format (”80qq”).)

qq: 00h to 31h in hexadecimal (external character number from 00 to 49

| External character number | Command value (80qq) |
|---|---|
| 00 | 8000 |
| 01 | 8001 |
| 02 | 8002 |
| . | . |
| . | . |
| . | . |
| 10 | 800A |
| 11 | 800B |
| 12 | 800C |
| 13 | 800D |
| 14 | 800E |
| 15 | 800F |
| 16 | 8010 |
| 17 | 8011 |
| . | . |
| . | . |
| . | . |
| . | . |
| . | . |
| . | . |
| 38 | 8026 |
| 39 | 8027 |

Example: Command for displaying the external character (20th character in external character memory) registered in Example (1) above

G243 X___ Y___ “8014” ;

**NOTE**

Be sure to specify a hexadecimal value from 00h to 27h for qq. Otherwise, the display function does not function normally.)

# 6.33 EXECUTION MACRO CALL MASK FUNCTION

## 6.33.1 Function

An execution macro call can be masked by using the executor parameter or macro variable.

The following execution macro calls can be masked:

Axis address call

T code macro call

T code subprogram call

A masked call instruction is handled as either an axis move command or a T code output command.

Examples) When bit 0 (AX1CL) of compilation parameter No. 9005 is set to 1 and bit 0 (MA1) of executor parameter No. 9010 is set to 0:

User program

```
00001 ;
G00 X10 Z10;  →  Calls execution macro program O9009.
G00 X20 Z20;  →  Calls execution macro program O9009.
     :
```

When bit 0 (AX1CL) of compilation parameter No. 9005 is set to 1 and bit 0 (MA1) of executor parameter No. 9010 is set to 1:

User program

```
00001 ;
G00 X10 Z10;  →  Moves to X10, Z10 in rapid traverse mode.
G00 X20 Z20;  →  Moves to X20, Z20 in rapid traverse mode.
     :
```

When bit 0 (TCAL) of compilation parameter No. 9002 is set to 1 and macro variable #8691 is set to 1:

```
<User program>                               <P–CODE program>
  00001 ;                                      09000 ;
  G00 X10 Z10 ;                                G00 X....
  T11 ; →Calls P-CODE program 09000               :
  G00 X20 Z20 ;                                #8691 = 1 ;
     :                                            :
  T12 ; →T12 code                              M99 ;
     :
```

## 6.33.2 Macro Variables

| Variable number | Function | Conversa-tional | Auxiliary | Execution |
|---|---|---|---|---|
| #8690 | Execution macro call mask function variable 1 | R/W | R/W | R/W |
| #8691 | Execution macro call mask function variable 2 | R/W | R/W | R/W |

R: Read enabled/W: Write enabled

#8690: Execution macro call mask function variable 1

1 = Masks a first axis address macro call.

2 = Masks a second axis address macro call.

4 = Masks a third axis address macro call.

8 = Masks a fourth axis address macro call.

16 = Masks a fifth axis address macro call.

32 = Masks a sixth axis address macro call.

64 = Masks a seventh axis address macro call.

128 = Masks an eighth axis address macro call.

#8691: Execution macro call mask function variable 2

1 = Masks a T code macro call.

One or more macro calls can be masked. For example, to mask the first and second axis address macro calls, specify variable #8690 as follows:

#8690 = 3 ;

**NOTE**

1 When data is written to variable #8690, the value of executor parameter No. 9010 also changes accordingly. When data is read from #8690, the value of executor parameter No. 9010 is also read. The same is true of variable #8691 and executor parameter No. 9011.

2 If data is written to variable #8690 or #8691 using a conversational macro (display macro) while an execution macro (machining macro) is executing, the write operation may be delayed.

# 7 Series 16/18 2–PATH DEDICATED CONTROL FUNCTION

The macro libraries for path 1 are shared as libraries for single–path control. The macro libraries for path 2 are dedicated. The macro libraries listed below are for the T series, and similar libraries are available for the M series.

Macro libraries for path 1

| | |
|---|---|
| 16–TA.EXEC/?? | : FANUC P–G Mark II/Mate |
| 18–TA.EXEC/?? | : FANUC P–G Mark II/Mate |
| F16TA_??.MEX | : 16–TA macro library for personal computers |
| F18TA_??.MEX | : 18–TA macro library for personal computers |
| F16TB_??.MEX | : 16–TB macro library for personal computers |
| F18TB_??.MEX | : 18–TB macro library for personal computers |
| F16TC_??.MEX | : 16–TC macro library for personal computers |
| F18TC_??.MEX | : 18–TC macro library for personal computers |
| F16TI_??.MEX | : 16*i*–TA macro library for personal computers |
| F18TI_??.MEX | : 18*i*–TA macro library for personal computers |

Macro libraries for path 2

| | |
|---|---|
| 16–TTA.SB.EXEC/?? | : FANUC P–G Mark II/Mate |
| 18–TTA.SB.EXEC/?? | : FANUC P–G Mark II/Mate |
| F16TTS??.MEX | : 16–TA macro library for personal computers |
| F18TTS??.MEX | : 18–TA macro library for personal computers |
| F16TTB??.MEX | : 16–TB macro library for personal computers |
| F18TTB??.MEX | : 18–TB macro library for personal computers |
| F16TTC??.MEX | : 16–TC macro library for personal computers |
| F18TTC??.MEX | : 18–TC macro library for personal computers |

Both path 1 and path 2 allow the coding of execution, conversational, and auxiliary macro programs.

With Series 16*i*/18*i*, common macro libraries are used for path 1 and path 2. The path to be selected is determined by making a selection with the keyword SYSTEM in the link control file. For details, refer to the ”FAPT MACRO COMPILER (For Personal Computer) PROGRAMMING MANUAL (B–66102E).”

The special 2–path control functions are explained below.

## 7.1 COMMON CONVERSATIONAL MACRO SCREEN

When macro executor ROM are mounted on tool posts 1 and 2, the conversational macro of the macro executor of tool post 1 can be executed, regardless of which tool post selects the conversational macro screen.

Set compilation parameter TTDSP (No. 9007, #0) to 1 for the executor of tool post 2. Set parameters 9038, 9040, and 9041 to the same values as in tool post 1. The conversational macro program of tool post 1 is executed even when the CUSTOM key is pressed while tool post 2 is being selected.

If the parameter is set to 0, the conversational macro programs selected by the heads of tool posts 1 and 2 are executed separately.

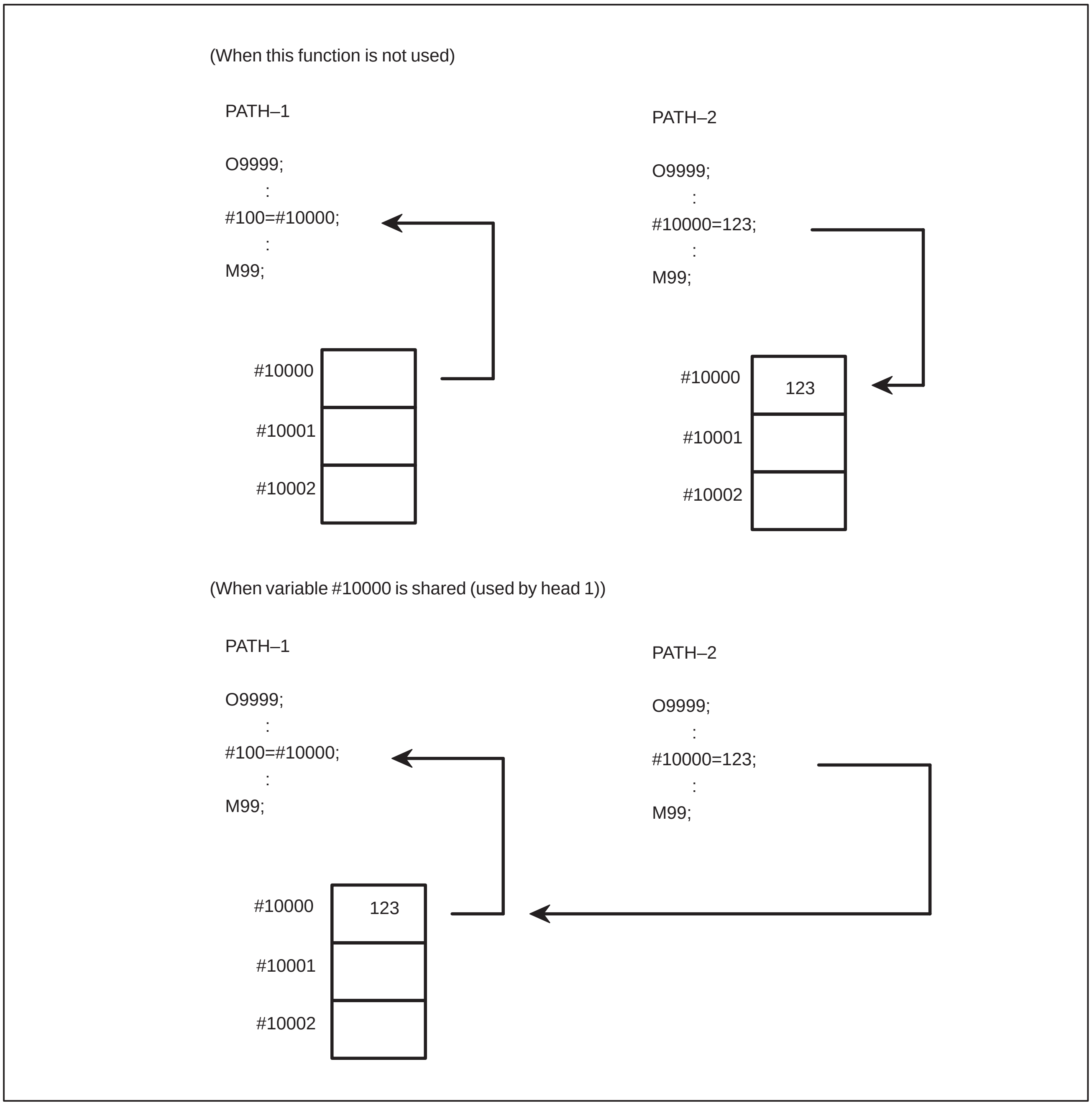
TTDSP=1 : The common conversational macro screen is validated.

**NOTE**
This function cannot be executed when the function for leaving the screen unchanged when the CUSTOM key is pressed (Section 6.26) is executed.

## 7.2 COMMON CONVERSATIONAL MACRO VARIABLE

When the macro executor ROM are mounted on tool posts 1 and 2, the heads of tool posts 1 and 2 can share conversational macro variables (#10000 and on and #20000 and on). When this function is validated, the execution, auxiliary, and conversational macro programs of tool posts 1 and 2 can share the conversational macro variables.

If this function is not used, heads 1 and 2 can use different conversational macro variables.



# 8 FUNCTIONS FOR STOPPING A CONVERSATIONAL MACRO

The macro executor can stop the execution of a conversational macro program at a particular program and sequence specified by parameters.

Parameters of the macro executor

1) No. 9000, bit 2 (TSTP)

TSTP 1 : The execution of the conversational macro program is stopped.

0 : The conversational macro program is executed.

2) No. 9002 : Number of the program at which execution of the conversational macro program is stopped

3) No. 9003 : Number of the sequence at which execution of the conversational macro program is stopped

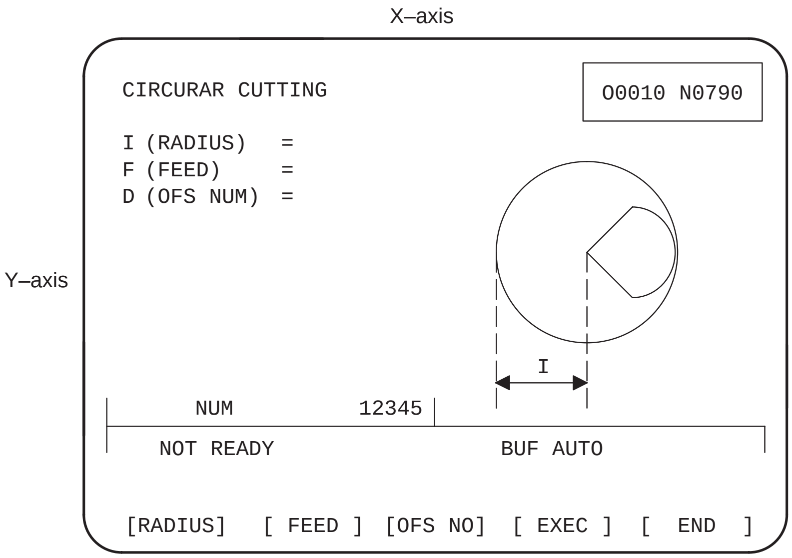
Specify the numbers of the program and sequence at which the conversational macro program is to be stopped and execute the conversational macro program on the CUSTOM screen. When an attempt is made to execute the block corresponding to the specified program and sequence, TSTP is automatically turned on. The execution of the conversational macro is stopped. If parameter No. 9002 is set to 0, this function is invalidated. In usual operation, it must be set to 0.

When this function is used, compilation parameter SEQ must be set to 1 to output sequence numbers to the P-CODE program.

# APPENDIX

# A MACRO PROGRAM EXAMPLE

# A.1 EXAMPLE–1: CIRCULAR CUTTING INPUT AND EXECUTION




Circular Cutting Display Layout:
The cursor is moved, using the cursor key ↑, or soft keys “RADIUS”, “FEED” or “OFSNUM”. Key in data from MDI at the cursor position and press INPUT to write data. Or after keying in data, press the soft key for direct input.
After all data has been input, press the soft key “EXEC” : the axis will move.
By pushing the soft key “END”, the display returns to the menu screen.

## A.1.1 Source Program for Main Program

```
O9999 ;
N001 (O0001 -- MAIN PROG -MENU           ) ;
N002 (O0010 -- KEY TEST                  ) ;
N003 (O0011 --    SUB SOFT KEY DISP      ) ;
N004 (O0020 -- CURSOR TEST               ) ;
N005 (O0030 -- DISPLAY TEST              ) ;
N006 (O0031 --    SUB TIMER              ) ;
N007 (O0032 --    SUB BLINK TEST         ) ;
N008 (O0033 --    SUB STRING DATA        ) ;
N009 (O0040 -- ADDRESS VARIABLE TEST     ) ;
N010 (O0041 --    SUB DATA TABLE         ) ;
N011 (O0050 -- CIRCLE TEST               ) ;
N012 (O0051 --    SUB DATA DISP          ) ;
N013 (O0052 --    SUB STRING DISP        ) ;
N014 (O0053 --    SUB GRAPHIC DISP       ) ;
N015 (O0054 --    SUB PMC WINDOW         ) ;
N016 (O9010 -- EXEC MACRO PROGRAM -G100- ) ;
 ;
N101 (V140 --- PROGRAM NUMBER            ) ;
```

```
N102 (V141 --- TIMER                        ) ;
N103 (V142 --- KEY CONTROL DATA   -V8501- ) ;
N104 (V143 --- STRINGS DATA OFFSET        ) ;
N105 (V144 --- KEY INPUT DATA     -V8503- ) ;
N105 (V145 --- ADDRESS INPUT DATA -V8504- ) ;
N107 (V146 --- CURSOR X POINTER           ) ;
N108 (V147 --- CURSOR Y POINTER           ) ;
N109 (V148 ---                            ) ;
N110 (V149 ---                            ) ;
 ;
 ;
 ;
 ;
 ;
 ;
```

```
O0001 ;
N000 G202 P3 ;
     #8505=0 ;
     G243 X0 Y0 A1 B0 (SAMPLE PROGRAM) ;
     X2 Y3 (1 -- KEY TEST) ;
     X2 Y5 (2 -- CURSOR TEST) ;
     X2 Y7 (3 -- DISPLAY TEST) ;
     X2 Y9 (4 -- ADDRESS VARIABLE) ;
     X2 Y11 (5 -- CIRCLE) ;
     #8509=0033 ;
     #143=300 ;
          M98 P0011 ;
N001 #142=#8501 ;
     IF [#142 EQ 0]  GOTO 99 ;
     IF [#142 LT 12] GOTO 99 ;
     IF [#142 GT 16] GOTO 99 ;
     #8500=[#142–11] *10 ;
     M99
N099 M99 P1 ;
 ;
 ;
 ;
 ;
 ;
```

### A.1.2 Source Program for Input Control

```
O0010 ;
     G202 P3 ;
     G243 X0 Y2 A1 B0 (KEY TEST   -- HIT ANY KEY --) ;
     #143=100 ;
          M98 P0011 ;
N001 #8502=2 ;
     #142=#8501 ;
     IF [#142 EQ 0] GOTO 99 ;
     #101=#142 ;
     #102=#8503 ;
     #103=#8504 ;
     G243 X0 Y4 A1 B0 (CONTROL  ) F8.3 Z0 D#101 ;
     G243 X0 Y5 A1 B0 (ADDRESS  ) F8.3 Z0 D#103 ;
     G243 X0 Y6 A1 B0 (DATA     ) F8.3 Z0 D#102 ;
N099 IF [#142 NE 16] GOTO 97 ;
N098 #8500=1 ;
     M99 ;
N097 M99 P1 ;
 ;
 ;
 ;
 ;
 ;
```

```
O0011 ;
     #100=0 ;
     WHILE [#100 LT 5] DO 1 ;
          G243 X[#100*8+1] Y16 A1 B0 P[#143+#100] ;
          #100=#100+1 ;
     END 1 ;
     M99;
 ;
 ;
 ;
 ;
 ;
```

## A.1.3 Source Program for Cursor Control

```
O0020 ;
          G202  P3 ;
          #8505=0 ;
          #8502=0 ;
     N008 G243 X0 Y1 (CURSOR TEST  –– MOVE CURSOR ––) ;
          G243 X0 Y3 (A) ;
          G243 X0 Y4 (B) ;
          G243 X0 Y5 (C) ;
          G243 X0 Y6 (D) ;
          G243 X0 Y7 (E) ;
          G243 X0 Y8 (F) ;
          #143=100 ;
               M98 P0011 ;
          #100=0 ;
          #8506=2 ;
     N001 #142=#8501 ;
          IF [#142 EQ 0] GOTO 3 ;
          IF [#142 NE 3] GOTO 2 ;
          #100=#100+1 ;
     N002 IF [#142 NE 4] GOTO 3 ;
          #100=#100-1 ;
     N003 #100=#100+6 ;
          #100=#100–[FI×[#100/6]]*6 ;
          #8507=#100+3 ;
          #8505=1 ;
     N099 IF [#142 NE 16] GOTO 97 ;
     N098 #8500=1 ;
          M99 ;
     N097 M99 P1 ;
 ;
 ;
 ;
 ;
 ;
```

## A.1.4 Source Program for Character Display

```
O0030 ;
     G202  P3 ;
     #8502=0 ;
     #8505=0 ;
     #143=100 ;
          M98 P0011 ;
N001 G243 X0 Y2 A1 B0 K200 ;
     X0 Y2 (DISPLAY TEST) ;
     G243 X0 Y4 A1 B0 K200 ;
     #141=100 ;
          M98 P0031 ;
     X0 Y4 (FANUC 0 SERIES MACRO COMPILER) ;
     #141=300 ;
          M98 P0031 ;
     G243 X0 Y4 A1 B0 K200 ;
     #141=100 ;
          M98 P0031 ;
     G243 X0 Y4 A1 B0 K200 ;
     X0 Y4 (*3441 2438 493D 3C28 00C3 00BD 00C4*) ;
     #141=300 ;
         M98 P0031 ;
     G243 X0 Y4 A1 B0 K200 ;
     #141=100 ;
          M98 P0031 ;
     G243 X0 Y4 A1 B0 K200 ;
     X0 Y4 (INT. CODE   ) (*2F40 2F48 2F79 2F53*) ;
     #141=300 ;
          M98 P0031 ;
     G243 X0 Y4 A1 B0 K200 ;
     #141=100 ;
          M98 P0031 ;
     X0 Y4 A3 B0 (3 MULTI) ;
     #141=300 ;
          M98 P0031 ;
     G243 X0 Y4 A1 B0 K200 ;
     #141=100 ;
          M98 P0031 ;
          M98 P0032 ;
     #141=100 ;
```

```
          M98 P0031 ;
     G243 X0 Y4 A1 B0 K200 ;
     X0 Y4 (DATA DISPLAY TEST     1234.567) ;
     #100=1234.567 ;
     G243 X0 Y6 F8.3 Z0 K200 ;
     X0 Y6 (F8.3  ) D#100 ;
     #141=300 ;
          M98 P0031 ;
     G243 X0 Y6 F8.3 Z1 K200 ;
     #141=100 ;
          M98 P0031 ;
     X0 Y6 (F8.3  ) D#100 (  LEADING ZERO NEG.) ;
     #141=300 ;
        M98 P0031 ;
     G243 X0 Y6 F5.2 K200 ;
     #141=100 ;
          M98 P0031 ;
     X0 Y6 (F5.2  ) D#100 ;
     #141=300 ;
          M98 P0031 ;
     #8509=0033 ;
     G243 X0 Y4 A1 B0 K200 ;
     #141=100 ;
         M98 P0031 ;
     X0 Y4 (STRINGS DISPLAY TEST) ;
     G243 X0 Y6 K200 ;
     X0 Y6 P10 ;
     #141=300 ;
          M98 P0031 ;
     G243 X0 Y6 K200 ;
     #141=100 ;
          M98 P0031 ;
     X0 Y6 P20 ;
     #141=300 ;
          M98 P0031 ;
     #142=#8501 ;
N099 IF [#142 NE 16] GOTO 97 ;
N098 #8500=1 ;
     M99 ;
     M99 P1 ;
 ;
 ;
 ;
 ;
```

```
 ;
```

```
O0031 ;
N001 IF [#141 LT 0] GOTO 99 ;
     #141=#141–1 ;
     GOTO 1 ;
N099 #142=#8501 ;
     IF [#142 EQ 16] GOTO 97 ;
     M99 ;
N097 M99 P98 ;
 ;
 ;
 ;
 ;
 ;
```

```
O0032 ;
     G243 X0 Y4 A1 B0 (BLINK TEST) ;
     #100=200 ;
          WHILE [#100 GT 0] DO 1 ;
               #100=#100–1 ;
N001           G243 X0 Y6 A1 B1 (BLINK SLOW) ;
          END 1 ;
     G243 X0 Y6 A1 B0 K20 ;
     #141=100 ;
          M98 P0031 ;
     #100=200 ;
          WHILE [#100 GT 0] DO 2 ;
               #100=#100–1 ;
N002           G243 X0 Y6 A1 B2 (BLINK FAST) ;
          END 2 ;
     G243 X0 Y4 A1 B0 K200 ;
     M99 ;
 ;
 ;
 ;
 ;
 ;
```

```
O0033 ;
N010 (ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789) ;
N020 (FANUC TECHNICAL TRAINING CENTER) ;
N100 (      ) ;
N101 (      ) ;
N102 (      ) ;
N103 (      ) ;
N104 ( END  ) ;
N200 (RADIUS) ;
N201 ( FEED ) ;
N202 (OFS NO) ;
N203 ( EXEC ) ;
N204 ( END  ) ;
N300 (TEST–1) ;
N301 (TEST–2) ;
N302 (TEST–3) ;
N303 (TEST–4) ;
N304 (TEST–5) ;
N500 (INPOSITION WIDTH   ) ;
N504 (SERVO ERROR LIMIT  ) ;
N508 (GRID SHIFT VALUE   ) ;
N512 (LOOP GAIN MULTIPLY ) ;
M99 ;
 ;
 ;
 ;
 ;
 ;
```

## A.1.5 Source Program for Address Variables

```
O0040 ;
     G202 P3 ;
     #143=100 ;
          M98 P0011 ;
N008 G243 X0 Y2 A1 B0 K520 ;
     G243 X0 Y2 A1 B0 (ADDRESS VARIABLE TEST) ;
     #141 =100 ;
           M98 P0031 ;
     X0 Y4 (ADDRESS G READ TEST) ;
N001 #101=G121.4 ;
     IF [#101 EQ 0] GOTO 2 ;
     X0 Y6 Z1 F1.0 (ESP STATUS –– ) D#101 (  PUSH ESP) ;
     #142=#8501 ;
     IF [#142 EQ 16] GOTO 98 ;
     M99 P1 ;
N002 X0 Y6 Z1 F1.0 (ESP STATUS –– ) D#101 (   RESET ESP) ;
     #141=100 ;
          M98 P0031 ;
     X0 Y4 K200 ;
     #141=100 ;
          M98 P0031 ;
     X0 Y4 (ADDRESS D WRITE TEST) ;
          M98 P0041 ;
     G243 X0 Y4 A1 B0 K200 ;
     #141=100 ;
          M98 P0031 ;
     G243 X0 Y4 (PARAMETER READ) ;
     #102=0 ;
     #8509=0033 ;
          WHILE [#102 LE 3] DO 1 ;
              #103=P[#102*4+500] ;
              G243 X0 Y[#102+6] (NUM ) Z1 F3.0 D[#102*4+500] ;
              ( –– ) D#103 ;
              G243 (  ) P[#102*4+500] ;
              #102=#102+1 ;
          END 1 ;
     #141=500 ;
          M98 P0031 ;
     #142=#8501 ;
```

```
N099 IF [#142 NE 16] GOTO 97 ;
N098 #8500=1 ;
     M99 ;
N097 M99 P8 ;
 ;
 ;
 ;
 ;
 ;
```

```
O0041 ;
     #100=0 ;
          WHITE [#100 LT 5] DO 1 ;
               G310 D699 Q#100 ;
               #141=10 ;
                    M98 P0031 ;
               #100=#100+1 ;
          END 1 ;
               #100=D699 ;
     G243 X0 Y6 (DATA TABLE –– ) F3.0 D#100 ;
     #141=300 ;
          M98 P0031 ;
     M99 ;
 ;
 ;
 ;
 ;
 ;
```

## A.1.6 Source Program for Graphic Display and Circle Cutting

```
O0050 ;
     G202 P3 ;
     #8506=13 ;
     #8507=2 ;
     #8505=1 ;
     #147=0 ;
     G202 P3 ;
          M98 P0052 ;
          M98 P0053 ;
          M98 P0051 ;
N001 #8502=1 ;
     #142=#8501 ;
     #144=#8503 ;
     IF [#142 EQ 0] GOTO 99 ;
     IF [#142 NE 12] GOTO 3 ;
     #147=0 ;
     GOTO 9 ;
N003 IF [#142 NE 13] GOTO 4 ;
     #147=1 ;
     GOTO 9 ;
N004 IF [#142 NE 14] GOTO 5 ;
     #147=2 ;
     GOTO 9 ;
N005 IF [#142 NE 4] GOTO 6 ;
     #147=#147–1 ;
     GOTO 9 ;
N006 IF [#142 NE 3] GOTO 7 ;
     #147=#147+1 ;
     GOTO 9 ;
N007 IF [#142 NE 15] GOTO 8 ;
     #140=1 ;
          M98 P0054 ;
     GOTO 10 ;
N008 IF [#142 NE 8] GOTO 10 ;
N009 #147=#147+3 ;
     #147=#147–FIX[#147/3]* 3 ;
     IF [#144 EQ #0] GOTO 10 ;
     #[500+#147]=#144 ;
     G243 X14 Y[#147+2] Z1 F4.0 D#[#147+500] ;
```

```
N010 M98 P0051 ;
N099 IF [#142 NE 16] GOTO 97 ;
N098 #8500=1 ;
     M99 ;
N097 M99 P1 ;
 ;
 ;
 ;
 ;
 ;
```

```
O0051 ;
     #8507=#147+2 ;
     #100=0 ;
       WHILE [#100 LT 3] DO 1 ;
            G243 X14 Y[#100+2] Z1 F4.0 D#[500+#100] ;
            #100=#100+1 ;
       END 1 ;
     M99 ;
 ;
 ;
 ;
 ;
 ;
```

```
O0052 ;
     G243 X0 Y0 A1 B0 (CIRCULAR CUTTING) ;
     X1 Y2 (I) ;
     C40 (PADIUS) C41 C61 ;
     X1 Y3 (F) ;
     C40 ( FEED ) C41 C61 ;
     X1 Y4 (D) ;
     C40 (OFS NO) C41 C61 ;
     #143=200 ;
          M98 P0011 ;
     M99 ;
 ;
 ;
 ;
 ;
 ;
```

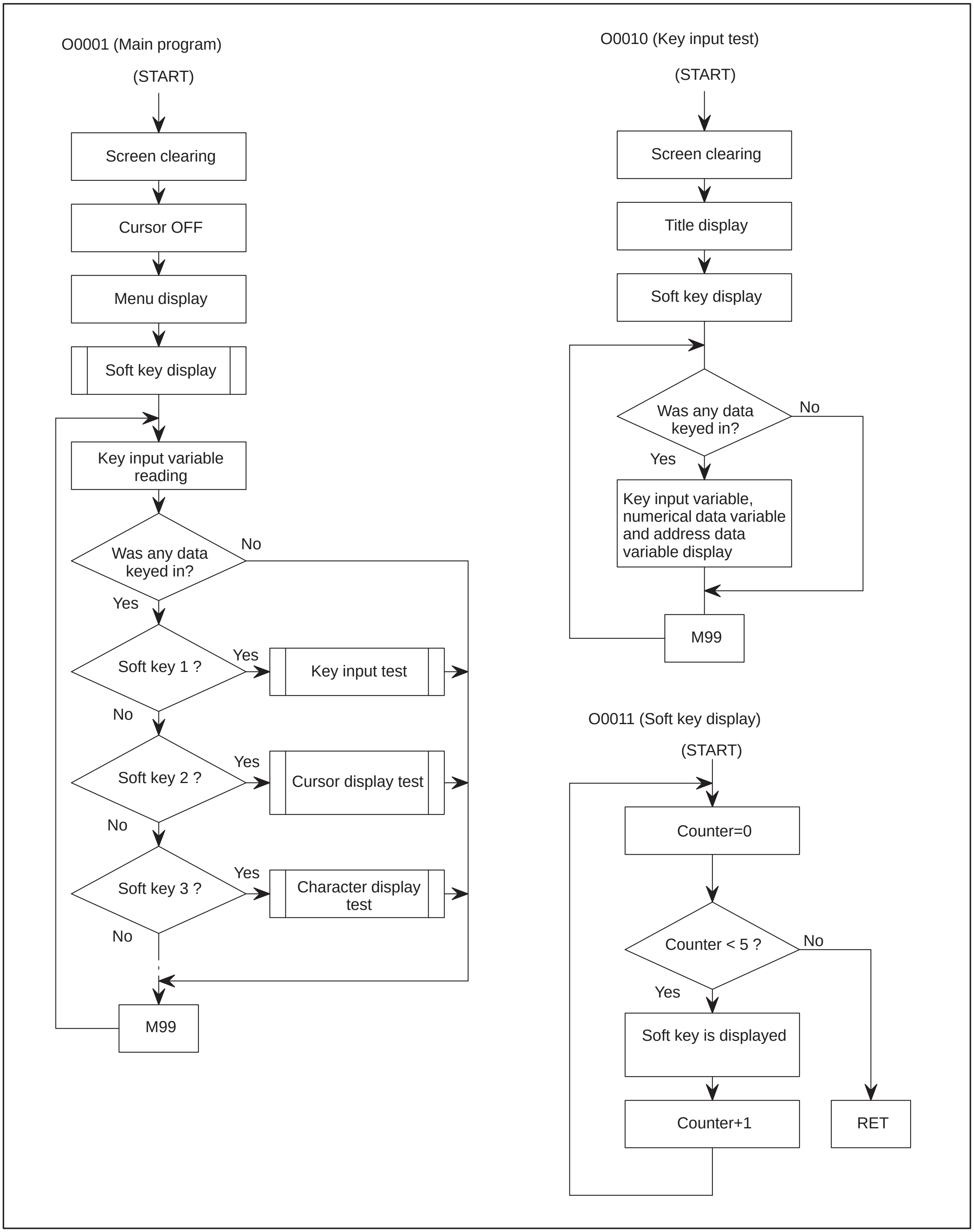
```
O0053 ;
     G244 P0 ;
     G242 X80 Y20 ;
     G01 X110 Y50 ;
     G02 X140 Y20 I110 J20 Q0 ;
     G02 X140 Y20 I80 J20 Q4 ;
     G02 X110 Y-10 I110 J20 Q0 ;
     G01 X80 Y20 ;
     G244 P2 ;
     G242 X80 Y20 ;
     G01 Y–70 ;
     G242 X20 Y20 ;
     G01 Y–70 ;
     G244 P1 ;
     G242 X80 Y-65 ;
     G01 X20 ;
     G243 X25 Y12 A1 B0 (I) ;
     M99 ;
 ;
 ;
 ;
 ;
 ;
```

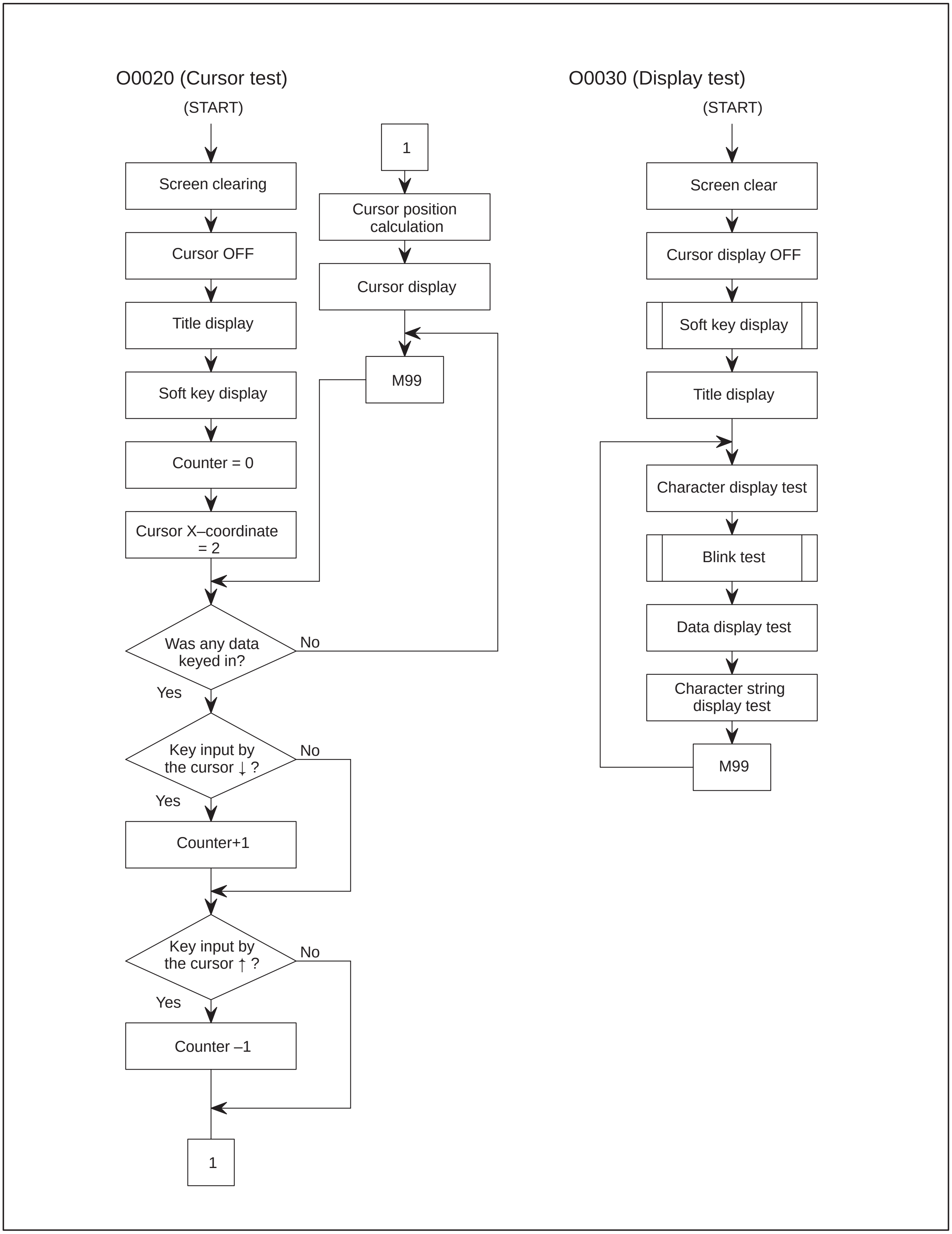
```
O0054 ;
     G310 D699 Q[#140+16] ;
     #141=5 ;
          M98 P0031 ;
     G310 D699 Q[#140+48] ;
     #141=5 ;
          M98 P0031 ;
     G310 D699 Q[#140+16] ;
     #141=5 ;
          M98 P0031 ;
     G310 D699 Q0 ;
     M99 ;
 ;
 ;
 ;
 ;
 ;
```

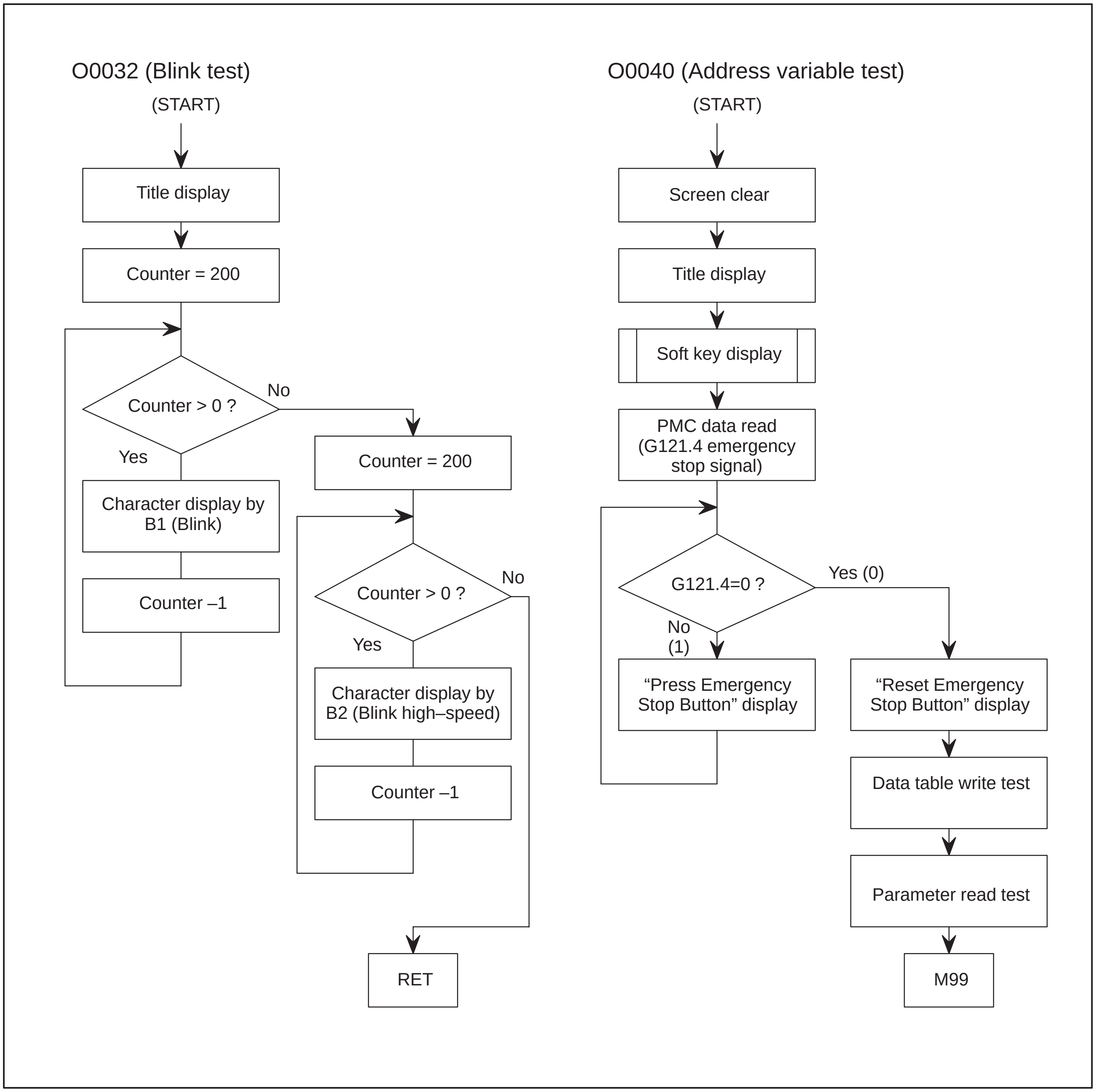
## A.1.7 Execution Macro

```
00020       G40 G49 G80 ;
00030       #1=#500/2 ;
00040       G91 G42 G01 X#1 Y#1 D#502 F#501 ;
00050       G02 X#1 Y–#1 J–#1 ;
00060       I–#500 ;
00070       X–#1 Y–#1 I–#1 ;
00080       G40 G01 X–#1 Y#1 ;
00090       M99 ;
00100   ;
00110   ;
00120   ;
00130   ;
00140   ;
```

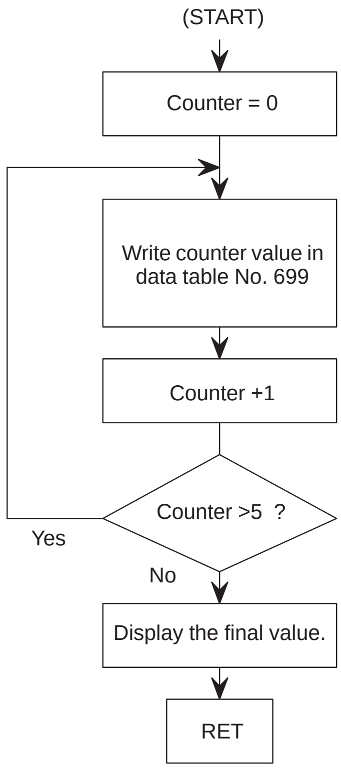
## A.1.8 Flow Chart



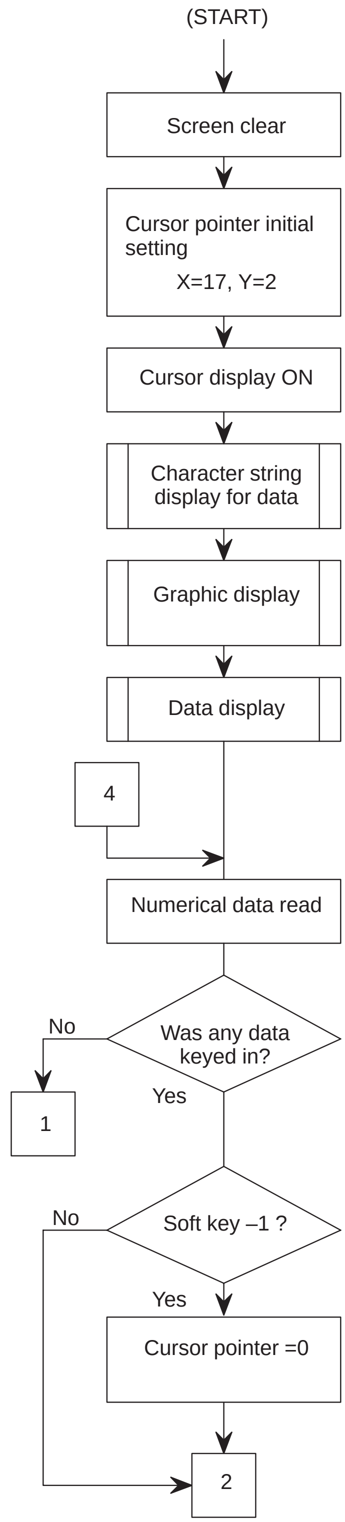

O0020 (Cursor test)
(START)
Screen clearing
Cursor OFF
Title display
Soft key display
Counter = 0
Cursor X–coordinate = 2
Was any data keyed in?
No
Yes
Key input by the cursor ↓ ?
No
Yes
Counter+1
Key input by the cursor ↑ ?
No
Yes
Counter –1
1
1
Cursor position calculation
Cursor display
M99
O0030 (Display test)
(START)
Screen clear
Cursor display OFF
Soft key display
Title display
Character display test
Blink test
Data display test
Character string display test
M99

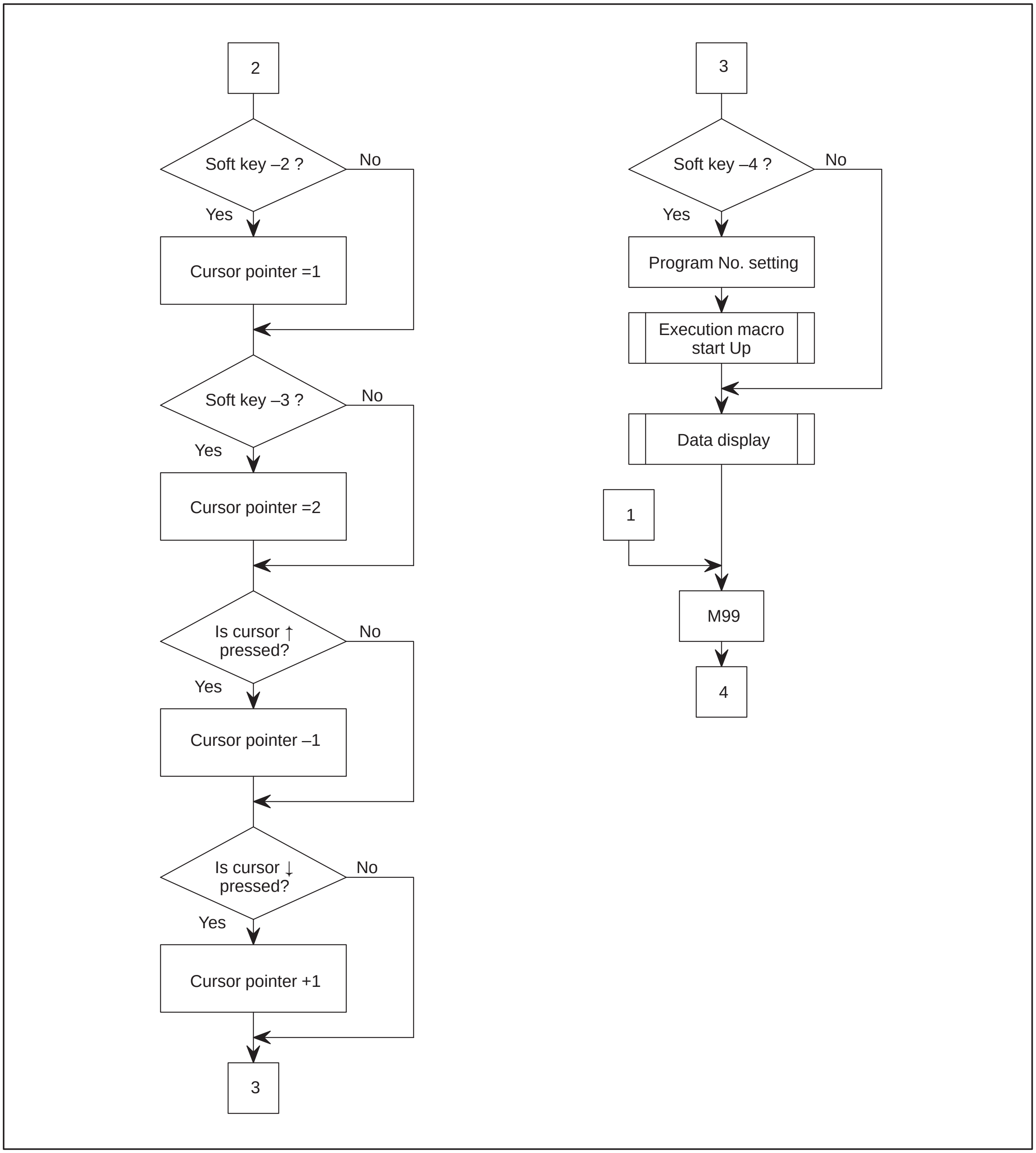

O0032 (Blink test)
(START)
Title display
Counter = 200
Counter > 0 ?
No
Yes
Character display by B1 (Blink)
Counter –1
Counter = 200
Counter > 0 ?
No
Yes
Character display by B2 (Blink high–speed)
Counter –1
RET
O0040 (Address variable test)
(START)
Screen clear
Title display
Soft key display
PMC data read (G121.4 emergency stop signal)
G121.4=0 ?
Yes (0)
No (1)
“Press Emergency Stop Button” display
“Reset Emergency Stop Button” display
Data table write test
Parameter read test
M99

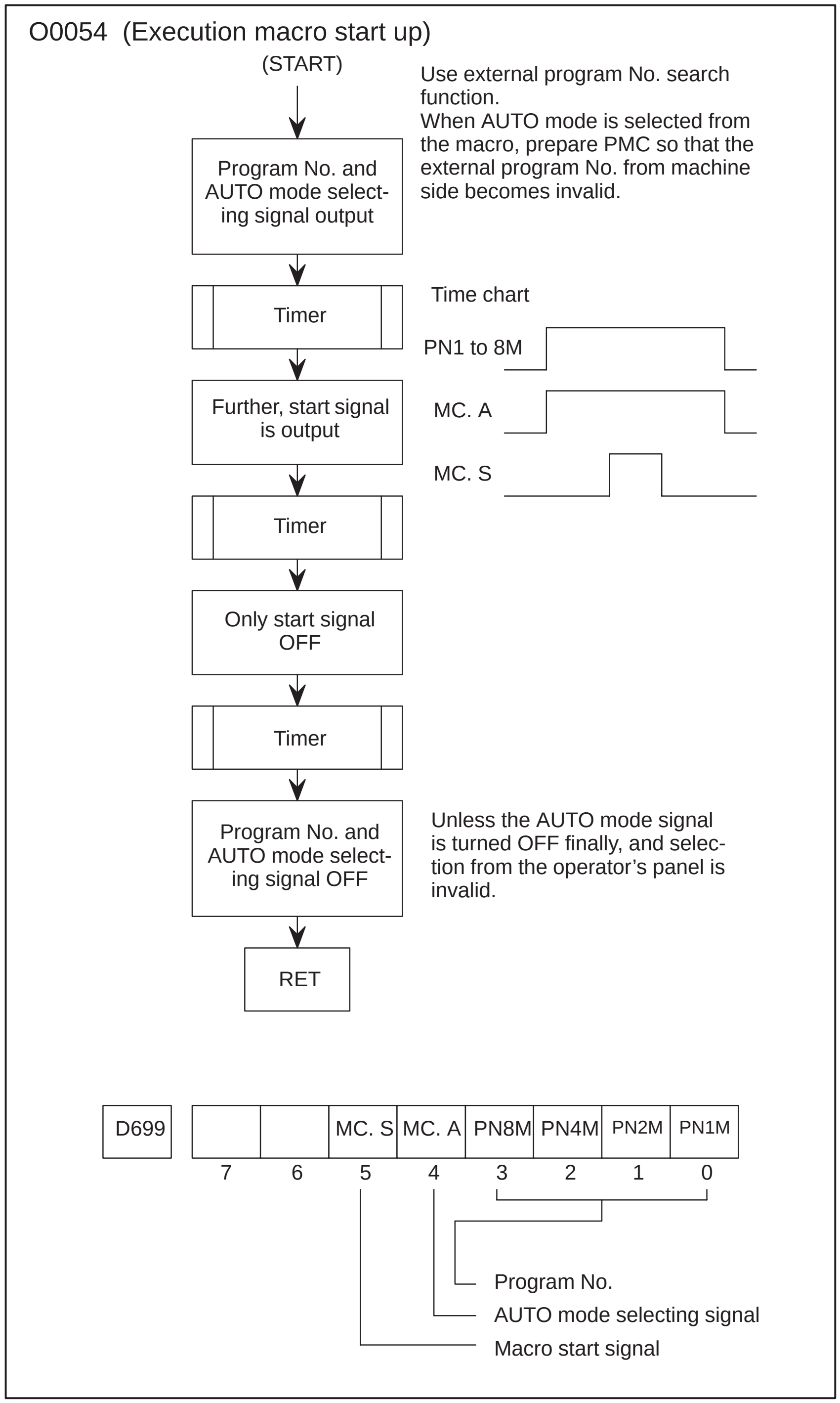
O0041 (Data table write test)




O0050 (Circular cutting)




2
Soft key –2 ?
No
Yes
Cursor pointer =1
Soft key –3 ?
No
Yes
Cursor pointer =2
Is cursor ↑ pressed?
No
Yes
Cursor pointer –1
Is cursor ↓ pressed?
No
Yes
Cursor pointer +1
3
3
Soft key –4 ?
No
Yes
Program No. setting
Execution macro start Up
Data display
1
M99
4


O0054 (Execution macro start up)
(START)
Use external program No. search function.
When AUTO mode is selected from the macro, prepare PMC so that the external program No. from machine side becomes invalid.
Program No. and AUTO mode selecting signal output
Timer
Time chart
PN1 to 8M
MC. A
MC. S
Further, start signal is output
Timer
Only start signal OFF
Timer
Program No. and AUTO mode selecting signal OFF
Unless the AUTO mode signal is turned OFF finally, and selection from the operator's panel is invalid.
RET
D699
MC. S
MC. A
PN8M
PN4M
PN2M
PN1M
7
6
5
4
3
2
1
0
Program No.
AUTO mode selecting signal
Macro start signal

## A.1.9 Explanation of Program

(1) O0001

0020 N000 G202 P3

Screen erase

P1: Character screen, P2: Graphic screen,
P3: Character and graphics

0030 #8505=0

Cursor control

#8505=0 : Cursor OFF, #8505=1 : Cursor ON

0040 G243 X0 Y0 A1 B0 (SAMPLE PROGRAM)

X2 Y3 (······)

Displaying the characters on the screen
G243 : Character display
X0 : Position to start X-axis display
Y0 : Position to start Y-axis display
A1 : Size of the letter (A1 : 1 time, A3 : 3 times)

| 0, 0 | 39, 0 |
|---|---|
| Character coordinate | |
| 0, 15 | 39, 15 |

The character string in parenthesis is displayed on the screen.

* In NC programming, a set of parentheses ( ) is a control IN/OUT function, which can be used as a comment, but in conversation macro, it means the display data, which cannot be used as a comment.

0100 #8509=0033

#143=300

M98 P0011

#8509 is a character string registeration control variable.

The character string corresponding to a sequence No. can be displayed, using G243 <u>XYP sequence No.</u>

Set Program No. including Sequence No., using #8509.

#143 is a common variable, which is used as offset of Sequence No. here.

The menu corresponding to the soft key is displayed, using subprogram O0011.

0130 N001 #142=#8501

#8501 is a key input variable, which accepts the cursor key, page key, soft key, edit key and input key.

Once it is read, it is cleared to ”0”;then, save the data in #142.

(2) O0011

0030 WHILE [Conditional Expression] D01

~

END1

In a WHILE statement, while the conditional expression is established, processing between DO and END is carried out, and when it is not established, execution is started from the next block of the corresponding END statement.

0040 G243 X [ ] Y [ ] P [#143+#100]

#143 is a common variable, which is used for Sequence No. offset.

#100 is used as a counter, which counts 0 to 4.

Here set the head Sequence No. of the character string written in Program No. O0033, using #143, and loop it with #100 to make it correspond to soft key 1 to 5.

<u>O0020</u>

0220 N003 #100=#100+6

#100=#100–[FIX [#100/6]]*6

This calculation is made to find the cursor position.

Add 6 in advance so that cursor position does not become negative on the 220th line. ”6” means that the cursor moves at 6 places.

”FIX” on the 230th line is a function which omits the figure below the decimal place of the answer for the quotient. Find the remainder divided by 6.

<u>O0030</u>

0090 G243 X0 Y2 A1 B0 K200

Write 200 blanks, using K200.

If a triple character is specified with A3, the blank of a triple character is written by the specified number.

0190 X0 Y4 (*3441 2438 493D......*)

When internal codes, such as Chinese characters are used for the screen display, enclose them with ”(*”and ”*)”.

For chinese characters, space for 2 letters are used.

0480 G243 X__Y__ F8.3

F8.3 specifies the digits when numerical data is displayed: total 8 digits and 3 digits below the decimal place.

0570 X6 Y6 (5.2)

#100=1234.567

When the display is made in 2 digits below the decimal place, using F5.2 format, the figure is half-adjusted to ”1234.57”.

0600 #8509=0033

Use the character string registeration program variable #8509 that designates program No. including sequence No.specified by G243 P(<u>Sequence No.</u>).

Here, use Program No. 0033 character string.

<u>O0032</u>

G243·····B1(···)

For blink display, display instructions are given only once but repeated.

<u>O0040</u>

0101 #101=G121.4

G121.4 (emergency stop) condition is read to common variable #100.

0310 #103 =P [#102∗4+1821].1

The contents of parameter specified by P [ ] is read to common variable #103.

Here, value of #102 changes from 0 to 2 and parameters 1821, 1825, 1829 are read.

1821.1 Reference counter

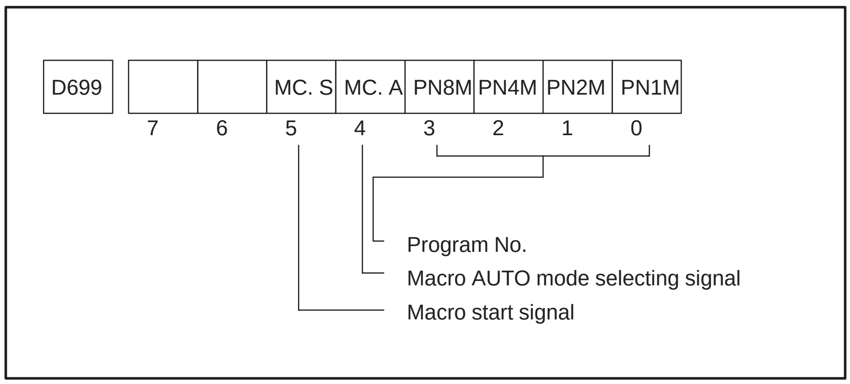
1825.1 Loop gain multiply

1829.1 Excessive error at stop

(3) O0054

Use data table No. 699 for data transfer with PMC, and write Program No.,

AUTO mode selection and cycle start signal.

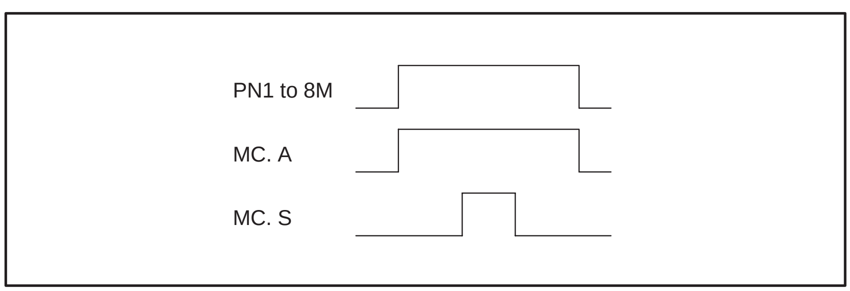



In the PMC sequence program, the external Program No. from the machine side was made invalid and Program No. from the macro valid when the AUTO mode from the macro (MC.A) is received.

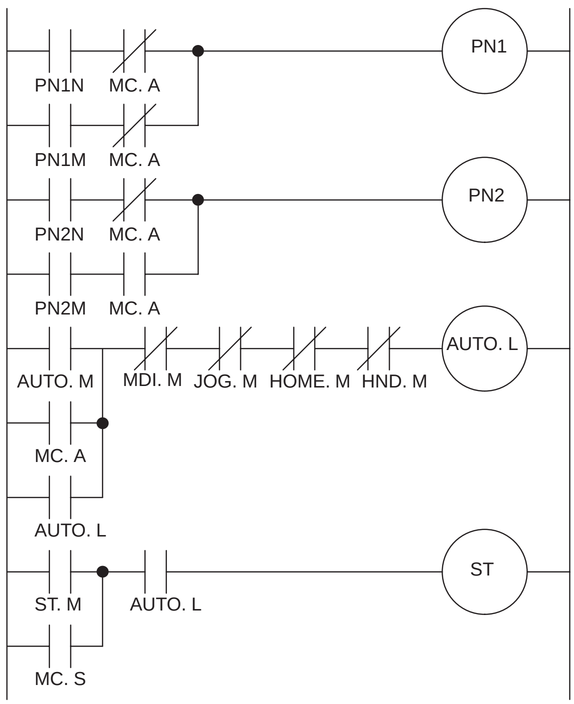
When MC.A signal is input, the mode on the operator’s panel is changed to ”AUTO” automatically.

When the timer preset time elapses after the mode selection, turn the cycle start signal ON.

Within PMC, this signal is ”OR” with the cycle start signal on the operator’s panel.



## A.1.10 Program Example for PMC Sequence




| | |
|---|---|
| MC. A : | AUTO mode selecting signal from MACRO |
| MC. S : | Cycle start signal from MACRO |
| PN1M : | External program No. from MACRO |
| PN1N : | External program No. from the machine |
| PN1 : | External program No. search output signal to NC |
| AUTO. M : | AUTO mode selecting signal from the operator's panel |
| MDI. M : | MDI mode selecting signal from the operator's panel |
| JOG. M : | JOG mode selecting signal from the operator's panel |
| HOME. M : | ZRN mode selecting signal from the operator's panel |
| HND. M : | HND mode selecting signal from the operator's panel |
| AUTO. L : | AUTO mode selecting signal(hold) |
| ST : | Cycle start signal to NC |
| ST. M : | Cycle start signal from the operator's panel |

```
O0001 ; Main
N000 G202 P3 ; Screen clear
     #8505=0 ; Cursor OFF
     G243 X0 Y0 A1 B0 (SAMPLE PROGRAM)
     X2 Y3 (1 -- KEY TEST) ;
     X2 Y5 (2 -- COUSOR TEST) ;           Menu display
     X2 Y7 (3 -- DISPLAY TEST) ;
     X2 Y9 (4 -- ADDRESS VARIABLE) ;
     X2 Y11 (5 -- CIRCLE) ;
     #8509=0033 ; Character string registration program No.
     #143=300 ;                           Soft key display
          M98 P0011 ;
N001 #142=#8501 ; Input variable read
     IF [#142 EQ 0] GOTO 99 ; Loop for which no key is depressed (including M99)
     IF [#142 LT 12] GOTO 99 ;
     IF [#142 GT 16] GOTO 99 ;
     #8500=[#142-11] *10 ;
     M99                                  Separate to program per softkeys
N099 M99 P1 ;
;
;
;
;
;
```

```
O0010 ; Key test
      G202 P3 ; Screen clear
      G243 X0 Y2 A1 B0 (KEYTEST    -- HIT ANY KEY --) ; Title display
      #143=100 ;          Soft key display: #143 is displayed at the
           M98 P0011 ;    head of character sequence No.
N001 #8502=2   Input an address and numeral
      #142=#8501 ; SAVE a key input variable
      IF [#142 EQ 0] GOTO 99 ;
      #101=#142 ;                   Key input presence or absence judgement,
      #102=#8503 ;                  loop for which key input is none.
      #103=#8504 ;  Key input data read
      G243 X0 Y4 A1 B0 (CONTROL  ) F8.3 Z0 D#101 ;
      G243 X0 Y5 A1 B0 (ADDRESS  ) F8.3 Z0 D#103 ;   Key input
      G243 X0 Y6 A1 B0 (DATA     ) F8.3 Z0 D#102 ;   data display
N099 IF [#142 NE 16] GOTO 97 ; Shift to the menu after a push
N098 #8500=1 ;         on the END key.
      M99 ;
N098 M99 P1 ;
;
;
;
;
;
```

```
O0011 ;
      #100=0 ; Clear the counter     Top sequence No. of
                                     character for soft key
      WHILE [#100 LT 5] DO 1 ;                        Counter
           G243 X[#100*8+1] Y16 A1 B0 P[#143+#100] ;
           #100=#100+1 ; Counter +1
      END 1 ;
      M99 ;
;
;
;
;
;
```

```
O0020 ; Cursor test
      G202  P3 ;
      #8505=0 ; Cursor display none
      #8502=0 ; key input none
N008 G243 X0 Y1 (CURSOR TEST  -- MOVE CURSOR --) ; Title display
      G243 X0 Y3 (A) ;
      G243 X0 Y4 (B) ;
      G243 X0 Y5 (C) ;
      G243 X0 Y6 (D) ;     The menu for cursor is displayed
      G243 X0 Y7 (E) ;
      G243 X0 Y8 (F) ;
      #143=100
           M98 P0011 ;    Soft key display
      #100=0 ; Pointer for cursor display posotion
      #8506=2 ; Cursor position in X-axis direction
N001 #142=#8501 ; SAVE key input control variable.
      IF [#142 EQ 0] GOTO 3 ; Key input presence or absence judgment
      IF [#142 NE 3] GOTO 2 ; Pointer +1 for cursor ↓
      #100=#100+1 ;
N002 IF [#142 NE 4] GOTO 3 ; Pointer –1 for cursor ↑
      #100=#100-1 ;
N003 #100=#100+6 ;
      #100=#100-[FIX [#100/6]]*6 ;    Pointer posotion calculation
      #8507=#100+3 ; Actual Y-axis cusor posotion
      #8505=1 ; Cursor display ON
N099 IF [#142 NE 16] GOTO 97 ; Return to the menu after a push on the END key.
N098 #8500=1 ;
      M99 ;
N097 M99 P1 ;
;
;
;
;
;
```

```
O0030 ; Character display test
      G202  P3 ; Screen clear
      #8502=0 ; Data input invalid
      #8505=0 ; Cursor display OFF
      #143=100 ;            } Soft key display
           M98 P0011 ;      }
N001 G243 X0 Y2 A1 B0 K200 ; Partial screen erase
      X0 Y2 (DISPLAY TEST) ;
      G243 X0 Y4 A1 B0 K200 ;
      #141=100 ;            } Timer
           M98 P0031 ;      }
      X0 Y4 (FANUC 0 SERIES  MACRO COMPILER) ;
      #141=300 ;                  ↖Enclose the characters to be displayed
           M98 P0031 ;              with a parenthesis.
      G243 X0 Y4 A1 B0 K200 ;
      #141=100 ;
           M98 P0031 ;
      G243 X0 Y4 A1 B0 K200 ;
      X0 Y4 (*3441 2438 493D 3C28 00C3 00BD 00C4*) ;
      #141=300 ;                        ↖Enclose the internal code with “(*”
           M98 P0031 ;                   and “*)”.
      G243 X0 Y4 A1 B0 K200 ;
      #141=100 ;
           M98 P0031 ;
      G243 X0 Y4 A1 B0 K200 ;
      X0 Y4 (INT.␣CODE␣)(*2F40␣2F48␣2F79 2F53*) ;
      #141=300 ;
           M98 P0031 ;
      G243 X0 Y4 A1 B0 K200 ;
      #141=100 ;
           M98 P0031 ;
      X0 Y4 A3 B0 (3 MULTI) ;
      #141=300 ;
           M98 P0031;
      G243 X0 Y4 A1 B0 K200 ;
      #141=100 ;
           M98 P0031 ;
           M98 P0032 ; Blink test sub program
      #141=100 ;
           M98 P0031 ;
      G243 X0 Y4 A1 B0 K200 ;
      X0 Y4 (DATA DISPLAY TEST      1234.567) ;
      #100=1234.567 ;
      G243 X0 Y6 F8.3 Z0 K200 ;
```

```
     X0 Y6 (F8.3  ) D#100 ;
     #141=300 ;
          M98 P0031 ;
     G243 X0 Y6 F8.3 Z1 K200 ;
     #141=100 ;
          M98 P0031 ;
     X0 Y6 (F8.3  ) D#100 (  LEADING ZERO NEG.) ;
     #141=300 ;
          M98 P0031 ;
     G243 X0 Y6 F5.2 K200 ;
     #141=100 ;
          M98 P0031 ;
     X0 Y6 (F5.2  ) D#100 ;
     #141=300 ;         Timer
          M98 P0031 ;
     #8509=0033 ; The character string is O0033.
     G243 X0 Y4 A1 B0 K200 ; Partial screen erase
     #141=100 ;         Timer
          M98 P0031 ;
     X0 Y4 (STRINGS DISPLAY TEST) ; Display
     G243 X0 Y6 K200 ; Screen partially erased
     X0 Y6 P10 ; The character string of sequence No. 10 is displayed.
     #141=300 ;         Timer
          M98 P0031 ;
     G243 X0 Y6 K200 ; Screen partially erased
     #141=100 ;         Timer
          M98 P0031 ;
     X0 Y6 P20 ; The character string of sequence No. 20 is displayed.
     #141=300 ;         Timer
          M98 P0031 ;
     #142=#8501 ; Key input variable is SAVED.
N099 IF [#142 NE 16] GOTO 97 ; Return to the menu when the soft key is “END”.
N098 #8500=1 ;
M99;
N097 M99 P1;
```

```
O0031 ; Timer
N001 IF [#141 LT 0] GOTO 99 ;      End when timer value is smaller than –10;
     #141=#141-1 ;
     GOTO 1 ;
N099 #142=#8501 ;
     IF [#142 EQ 16] GOTO 97 ;     Return to the menu when the END key
     M99 ;                         is depressed during timer operation
N097 M99 P98 ;
;
;
;
;
;
```

```
O0032 ; Brink test
     G243 X0 Y4 A1 B0 (BLINK TEST) ; Title display
     #100=200 ; Loop counter setting
          WHILE [#100 GT 0] DO 1 ;
               #100=#100-1 ;                            Loop for blinkig
N001           G243 X0 Y6 A1 B1 (BLINK SLOW) ;
          END 1 ;
     G243 X0 Y6 A1 B0 K20 ; Screen partially erased
     #141=100 ;                  Timer
          M98 P0031 ;
     #100=200 ; Loop counter setting
          WHILE [#100 GT 0] DO 2 ;
               #100=#100-1 ;                            Loop for blinking
N002           G243 X0 Y6 A1 B2 (BLINK FAST) ;
          END 2 ;
     G243 X0 Y4 A1 B0 K200 ; Screen partially erased
     M99 ;
   ;
   ;
   ;
   ;
   ;
```

```
O0033 ; Display character string data.
N010 (ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789) ;
N020 (FANUC TECHNICAL TRAINING CENTER) ;
N100 (      ) ;                  Enclose the display character string
                                 with parentheses.
N101 (      ) ;
N102 (      ) ;
N103 (      ) ;
N104 ( END  ) ;
N200 (RADIUS) ;
N201 ( FEED ) ;
N202 (OFS NO) ;    Enclose the internal code with “(*” and “*)”.
N203 ( EXEC ) ;
N204 ( END  ) ;
N300 (TEST-1) ;
N301 (TEST-2) ;
N302 (TEST-3) ;
N303 (TEST-4) ;
N304 (TEST-5) ;
N500 (INPOSITION WIDTH  ) ;
N504 (SERVO ERROR LIMIT ) ;
N508 (GRID SHIFT VALUE  ) ;
N512 (LOOP GAIN MULTPLY ) ;
M99 ;
  ;
  ;
  ;
  ;
  ;
```

```
O0040 ; Addreess variable test
      G202 P3 ; Screen erease
      #143=100 ;          } Soft key display
            M98 P0011 ;   }
N008 G243 X0 Y2 A1 B0 K520 ; Screen partially erased
      G243 X0 Y2 A1 B0 (ADDRESS VARIABLE TEST) ; Title display
      #141=100 ;          } Timer
            M98 P0031 ;   }
      X0 Y4 (ADDRESS G READ TEST) ; Display ;   Address G is
                                                PMC → NC input.
N001 #101=G121.4 ;   Emergency stop signal read
      IF [#101 EQ 0] GOTO 2 ;
                                               ↙Emergency stop signal status display
      X0 Y6 Z1 F1.0 (ESP STATUS -- ) D#101 (   PUSH ESP) ;
      #142=#8501 ;                  } Looped until the emergency stop signal is input,
      IF [#142 EQ 16] GOTO 98 ;     } but a push on the END key returns the display to
                                      the menu.
      M99 P1 ;
N002 X0 Y6 Z1 F1.0 (ESP STATUS -- ) D#101 (   RESET ESP) ;
      #141=100 ;          } Timer                 ↖Status display
            M98 P0031 ;   }
      X0 Y4 K200 ; Screen partialloy cleared
      #141=100 ;          } Timer
            M98 P0031 ;   }
      X0 Y4 (ADDRESS D WRITE TEST) ;
            M98 P0041 ; Subroutine of data table teat
      G243 X0 Y4 A1 B0 K200 ; Screen partially erased
      #141=100 ;          } Timer
            M98 P0031 ;   }
      G243 X0 Y4 (PARAMETER READ) ;
      #102=0 ; Counter
      #8509=0033 ;  The character string for display is O0033
            WHILE [#102 LE 2] DO 1 ;
                #103=P[#102*4+1821].1;Calculate parameter No.(P:parameter)
                G243 X0 Y[#102+6] (NUM ) Z1 F3.0 D[#102*4+1821]
                ( -- ) D#103 ;
                G243 (  ) P[#102*4+1821] ; “P” is sequence No. of character string.
                #102=#102+1
            END 1 ;
      #141=500 ;          } Timer
            M98 P0031 ;   }
_     #142=#8501 ;                  } A push on the END key returns
N099 IF [#142 NE 16] GOTO 97 ;      } the display to the menu.
N098 #8500=1 ;
      M99 ;
N097 M99 P8 ;
  ;
```

```
 ;
 ;
 ;
 ;
O0041 ;
     #100=0 ; Counter
          WHILE [#100 LT 5] DO 1 ;
               G310 D699 Q#100 ; Write Q data on the data table specified by D.
               #141=10 ;        } Timer
                    M98 P0031 ; }
               #100=#100+1 ; Counter +1
          END 1 ;
               #100=D699 ;                               } Answer display
     G243 X0 Y6 (DATA TABLE -- ) F3.0 D#100 ;  }
     #141=300 ;         } Timer
          M98 P0031 ;   }
     M99 ;
;
;
;
;
;
```

```
O0050 ; Circular cutting
      G202 P3 ;  Screen erased
      #8506=13 ; Cursor display X position
      #8507=2 ;  Cursor display Y position
      #8505=1 ;  Cursor display ON
      #147=0 ;   Cursor pointer
      G202 P3 ;
           M98 P0052 ; Character string display subroutine for data
           M98 P0053 ; Graphic display subroutine
           M98 P0051 ; Data display subroutine
N001 #8502=1 ; Data is input with numerals
      #142=#8501 ; SAVE key input variable.
      #144=#8503 ; Numeral data variable read
      IF [#142 EQ 0] GOTO 99 ; In the absence of key input, the loop M99 should
                               be included.
      IF [#142 NE 12] GOTO 3 ;
      #147=0 ;                    When soft key 1 is depressed, the cusor
                                  pointer is 0.
      GOTO 9 ;
N003 IF [#142 NE 13] GOTO 4 ;
      #147=1 ;                    When soft key 2 is depressed, the cursor
                                  pointer is 1.
      GOTO 9 ;
N004 IF [#142 NE 14] GOTO 5 ;
      #147=2 ;                    When the soft key 3 is depressed, the cursor
                                  pointeris 2.
      GOTO 9 ;
N005 IF [#142 NE 4] GOTO 6 ;
      #147=#147-1 ;               When the cursor is "↑", the pointer is –1.
      GOTO 9 ;
N006 IF [#142 NE 3] GOTO 7 ;
      #147=#147+1 ;               When the cursor is "↓", the pointer is +1.
      GOTO 9 ;
N007 IF [#142 NE 15] GOTO 8 ; To N8 if soft key 4 is not "EXEC".
      #140=1 ; Select O0001 external program seach No. → Select O0001 of NC memory.
           M98 P0054 ; Start NC via PMC.
      GOTO 10 ;
N008 IF [#142 NE 8] GOTO 10 ; To N10 if "INPUT" is not depressed.
N009 #147=#147+3
      #147=#147-FIX [#147/3]* 3 ; Cursor position calculation
      IF [#144 EQ #0] GOTO 10 ; To N10 if any numeric was kayed in.
      #[500+#147]=#144 ; Data written to common variable 500 –
      G243 X14 Y[#147+2] Z1 F4.0 D#[#147+500] ;
                                                  Data display
N010 M98 P0051 ;
N099 IF [#142 NE 16] GOTO 97 ;
                                  To the menu when the END key is depressed.
N098 #8500=1 ;
      M99 ;
N097 M99 P1 ;
```

```
;
;
;
;
;

O0051 ; Cursor and data display
     #8507=#147+2 ; Cusror and position
     #100=0 ; Counter
       WHILE {#100 LT 3] DO ! ;
            G243 X14 Y[#100+2] Z1 F4.0 D#[500+#100] ;    Data is displayed, changing
            #100=#100+1 ;                                display position.
       END 1 ;
     M99 ;
 ;
 ;
 ;
 ;
 ;

O0052 ; Character string display for data
     G243 X0 Y0 A1 B0 (CIRCULAR CUTTING) ;
     X1 Y2 (I) ;
     C40 (RADIUS) C41 C61 ;
     X1 Y3 (F) ;
     C40 ( FEED ) C41 C61 ;
     X1 Y4 (D) ;
     C40 (OFS NUM) C41 C61 ;
     #143=200 ;
          M98 P0011 ;
     M99 ;
 ;
 ;
 ;
 ;
 ;
```

```
O0053 ; Graphic display
     G244 P0 ; Type of line (solid line)
     G242 X80 Y20 ; Drawing start posotion
     G01 X110 Y50 ;
     G02 X140 Y20 I110 J20 Q0 ;
     G02 X140 Y20 I80 J20 Q4 ;
     G02 X110 Y-10 I110 J20 Q0 ;
                              ↖Specify the number of quadrants.
     G01 X80 Y20 ;
     G244 P2 ;
     G242 X80 Y20 ;
     G01 Y-70 ;
     G242 X20 Y20 ;
     G01 Y-70 ;
     G244 P1 ;
     G242 X80 Y-65 ;
     G01 X20 ;
     G243 X25 Y12 A1 B0 (I) ;
     M99 ;
 ;
 ;
 ;
 ;
 ;
```

```
00054 ; Data output to PMC
     G310 D699 Q[#140+16]; Select Program No. for #140 and AUTO mode with+16.
     #141=5 ;
          M98 P0031 ;       Timer
     G310 D699 Q[#140+48] ;AUTO mode and cycle start are ON with+48
     #141=5 ;
          M98 P0031 ;
     G310 D699 Q[#140+16] ;Only AUTO mode is ON with+16. Cycle start is OFF.
     #141=5 ;
          M98 P0031 ; All data including program No. is OFF.
     G310 D699 Q0 ;
     M99 ;
 ;
 ;
 ;
 ;
 ;
```

```
O9010 ; Execution macro for circular cutting
      G40 G49 G80 ;
      #1=#500/2 ;
      G91 G42 G01 X#1 Y#1 D#502 F#501 ;
      G02 X#1 Y-#1 J-#1 ;
      I-#500 ;
      X-#1 Y-#1 I-#1 ;
      G40 G01 X-#1 Y#1 ;
      M99 ;
 ;
 ;
 ;
 ;
 ;
```

Set 100 to compile parameter 9013, and call this program from program O0001 in NC memory, using G100. program O0001 is selected via PMC with external program No. search, from conversational macro O00054.

# A.2 EXAMPLE-2 EXAMPLE FOR KEY INPUT AND CURSOR CONTROL

## A.2.1 Source program List

```
O8000 ;
N1   M98 P8011 ;
N9   M98 P9503 ;
     IF [#102 EQ 12] GOTO 100 ;
     IF [#102 EQ 13] GOTO 200 ;
     M99 P9 ;
N100      M98 P8012 ;
N19       M98 P9503 ;
     IF [#102 EQ 16] GOTO 199 ;
     IF [#102 NE 8] GOTO 190 ;
     IF [#104 EQ 1] GOTO 110 ;
     #10050 = #103 ;
     #140 = 4.0 ;
     #141 = 1 ;
          M98 P9501 ;
     M99 P19 ;
N110 #[10051 + #105] = #103 ;
     #140 =5.2 ;
     #141 = 1 ;
          M98 P9501 ;
N190 M99 P19 ;
N199 M99 P1 ;
N200      M98 P8013 ;
N29       M98 P9503 ;
     IF [#102 EQ 16] GOTO 299 ;
     IF [#102 NE 8] GOTO 290 ;
     IF [#104 EQ 1] GOTO 210 ;
     #10060 = #103 ;
     #140 = 4.0 ;
     #141 = 0 ;
          M98 P9501 ;
     M99 P29 ;
N210 #[10061 + #105] = #103 ;
     #140 = 4.0 ;
```

```
     #141 = 0 ;
          M98 P9501 ;
N290 M99 P29 ;
N299 M99 P1 ;
 ;
 ;
 ;
 ;
 ;
```

```
O8001
     #10000 = 10010 ;
     #10001 = 2 ;
     #10002 = 10026 ;
     #10003 = 2 ;
     #10010 = 13 ;
     #10011 = 2 ;
     #10012 = 0 ;
     #10013 = 0 ;
     #10014 = 1 ;
     #10015 = 1 ;
     #10016 = 1 ;
     #10017 = 0 ;
     #10018 = 10 ;
     #10019 = 4 ;
     #10020 = 0 ;
     #10021 = 1 ;
     #10022 = 1 ;
     #10023 = 7 ;
     #10024 = 1 ;
     #10025 = 1 ;
     #10026 = 13 ;
     #10027 = 2 ;
     #10028 = 0 ;
     #10029 = 0 ;
     #10030 = 1 ;
     #10031 = 1 ;
     #10032 = 1 ;
     #10033 = 0 ;
     #10034 = 11 ;
     #10035 = 4 ;
     #10036 = 6 ;
     #10037 = 1 ;
     #10038 = 4 ;
```

```
      #10039 = 5 ;
      #10040 = 1 ;
      #10041 = 0 ;
      #8500 = 8000 ;
      M99 ;
  ;
  ;
  ;
  ;
  ;

O8011 ;
          M98 P9506 ;
      G243 X0 Y1 A1 (MACRO EXAMPLE) ;
      X2 Y4 A1 (SOFTKEY SELECT) ;
      X3 Y6 (1 - EXAMPLE A) ;
      X3 Y7 (2 - EXAMPLE B) ;
      X3 Y8 (3 - EXAMPLE C) ;
      X3 Y9 (4 - EXAMPLE D) ;
      X3 Y10(5 - EXAMPLE E) ;
      X2 Y16(EX.A) ;
      X10(EX.B) ;
      X18(EX.C) ;
      X26(EX.D) ;
      X34(EX.E) ;
      #100 = 0 ;
      M99 ;
  ;
  ;
  ;
  ;
  ;

O8012 ;
          M98 P9506 ;
      G243 X1 Y0 A1 (EXAMPLE A) ;
      X3 Y2 ( PROG NO ) C61 ;
      X0 Y4 (  DATA 1 ) C61 ;
      X0 Y5 (  DATA 2 ) C61 ;
      X0 Y6 (  DATA 3 ) C61 ;
      X0 Y7 (  DATA 4 ) C61 ;
      X0 Y8 (  DATA 5 ) C61 ;
      X0 Y9 (  DATA 6 ) C61 ;
      X0 Y10(  DATA 7 ) C61 ;
```

```
     X34 Y16 (END) ;
     G243 X14 Y2 F4.0 Z0 D#10050 ;
     #140 = 5.2 ;
     #141 = 1 ;
     #142 = 11 ;
     #143 = 4 ;
     #144 = 10051 ;
     #145 = 7 ;
     #146 = -9999 ;
     #147 = 1 ;
          M98 P9505 ;
     #100 = 1 ;
     #104 = 0 ;
     #105 = 0 ;
     #140 = 0 ;
          M98 P9502 ;
     M99 ;
 ;
 ;
 ;
 ;
 ;
```

```
O8013 ;
          M98 P9506 ;
     G243 X1 Y0 A1 (EXAMPLE B) ;
     X3 Y2 (PROG NO) (* 3D *) ;
     X0 Y4 (DATA SET1) ;
     X0 Y5 (DATA SET2) ;
     X0 Y6 (DATA SET3) ;
     X0 Y7 (DATA SET4) ;
     X0 Y8 (DATA SET5) ;
     X34 Y16 (END) ;
     G243 X14 Y2 F4.0 Z0 D#10060 ;
     #110 = 0 ;
     WHILE [#110 LE 4] D0 1 ;
          #140 = 4.0 ;
          #141 = 0 ;
          #142 = 12 ;
          #143 = #110 + 4 ;
          #144 = 10061 + #110 * 4 ;
          #145 = 4 ;
          #146 = -9999 ;
          #147 = 6 ;
```

```
                  M98 P9504 ;
             #110 = #110 + 1 ;
        END 1 ;
        #100 = 2 ;
        #104 = 0 ;
        #105 = 0 ;
        #140 = 0 ;
             M98 P9502 ;
        M99 ;
    ;
    ;
    ;
    ;
    ;
    ;
```

## A.2.2 Specification

### A.2.2.1 Type of Screen

(1) Initial screen

```
                 X-axis
 0
 1   MACRO EXAMPLE
 2
 3
 4   SOFTKEY SELECT

     1 – EXAMPLE A
     2 – EXAMPLE B          Y–axis
     3 – EXAMPLE C
 9   4 – EXAMPLE D
     5 – EXAMPLE E




14


  [ EX.A ]  [ EX.B ]  [ EX.C ]  [ EX.D ]  [ EX.E ]
```

Press CUSTOM key: this screen will be displayed.

On this screen, select a menu.

However, in this example, EX.C, EX.D, and EX.E menu screens are not prepared.

(2) Menu-1 (EX.A)

```
                 X-axis
 0   EXAMPLE A
 1
 2     PROG NO  –  _9999
 3
 4   DATA 1  = _999.99 _
     DATA 2  = _999.99
     DATA 3  = _999.99          Y–axis
     DATA 4  = _999.99
     DATA 5  = _999.99
 9   DATA 6  = _999.99
     DATA 7  = _999.99




14


  [     ]  [     ]  [     ]  [     ]  [ END ]
```

Press soft key **END** : the display will return to the initial screen.

”__” indicates the cursor position.

Actually, the cursor is displayed always at one place.

(3) Menu-2 (EX.B)

X-axis

Y–axis

```
0    EXAMPLE A
1
2      PROG NO  –  _9999
3
4    DATA SET1  _9999 _9999 _9999 _9999
     DATA SET2  _9999 _9999 _9999 _9999
     DATA SET3  _9999 _9999 _9999 _9999
     DATA SET4  _9999 _9999 _9999 _9999
     DATA SET5  _9999 _9999 _9999 _9999
9



14

      [     ]  [     ]  [     ]  [     ]  [ END ]
```

Press soft key END : the display will return to the initial screen.
”__” indicates the cursor position.
Acutally, the cursor is displayed always at one place.

## A.2.2.2 Variable Data

(1) Cursor control data

P-CODE variable

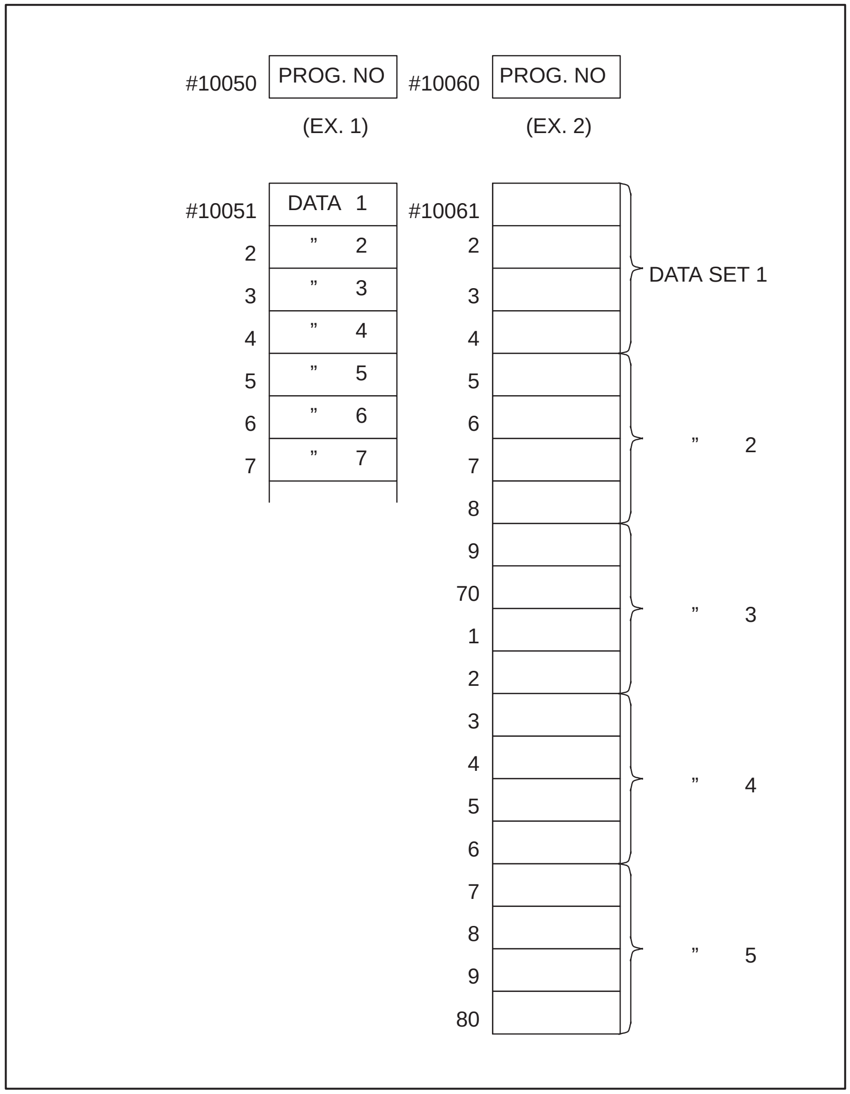
| | P-CODE variable | |
|---|---|---|
| #100 | CSNO | Cursor control No. |
| 1 | | |
| 2 | KCONT | Key input control data (#8501 value) |
| 3 | KDATA | Key input numerical data (#8503 value) |
| 4 | CPNT | Cursor pointer |
| 5 | SUBP | Cursor subpointer |

(2) Cursor data

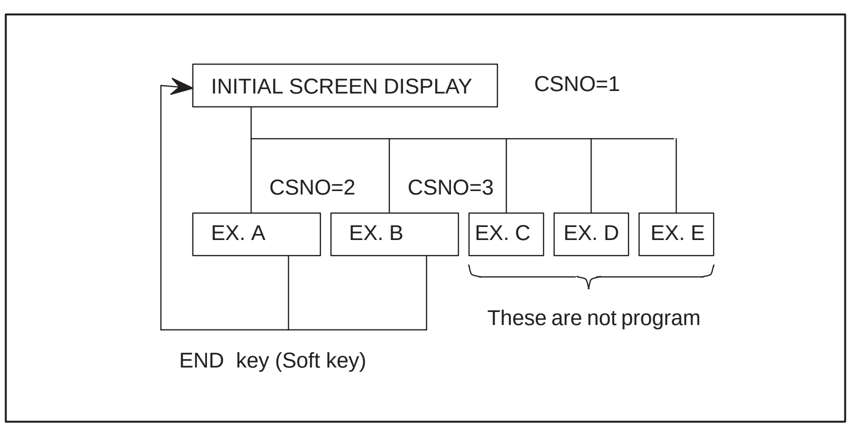
| | |
|---|---|
| #10000 | 10010 |
| | 2 |
| | 10026 |
| | 2 |

| | |
|---|---|
| #10010 | 13 |
| 1 | 2 |
| 2 | 0 |
| 3 | 0 |
| 4 | 1 |
| 5 | 1 |
| 6 | 1 |
| 7 | 0 |
| 8 | 10 |
| 9 | 4 |
| 20 | 0 |
| 1 | 1 |
| 2 | 1 |
| 3 | 7 |
| 4 | 1 |
| 5 | 1 |
| #10026 | 13 |
| 7 | 2 |
| 8 | 0 |
| 9 | 0 |
| 30 | 1 |
| 1 | 1 |
| 2 | 1 |
| 3 | 0 |
| 4 | 11 |
| 5 | 4 |
| 6 | 6 |
| 7 | 1 |
| 8 | 4 |
| 9 | 5 |
| 40 | 1 |
| 1 | 0 |

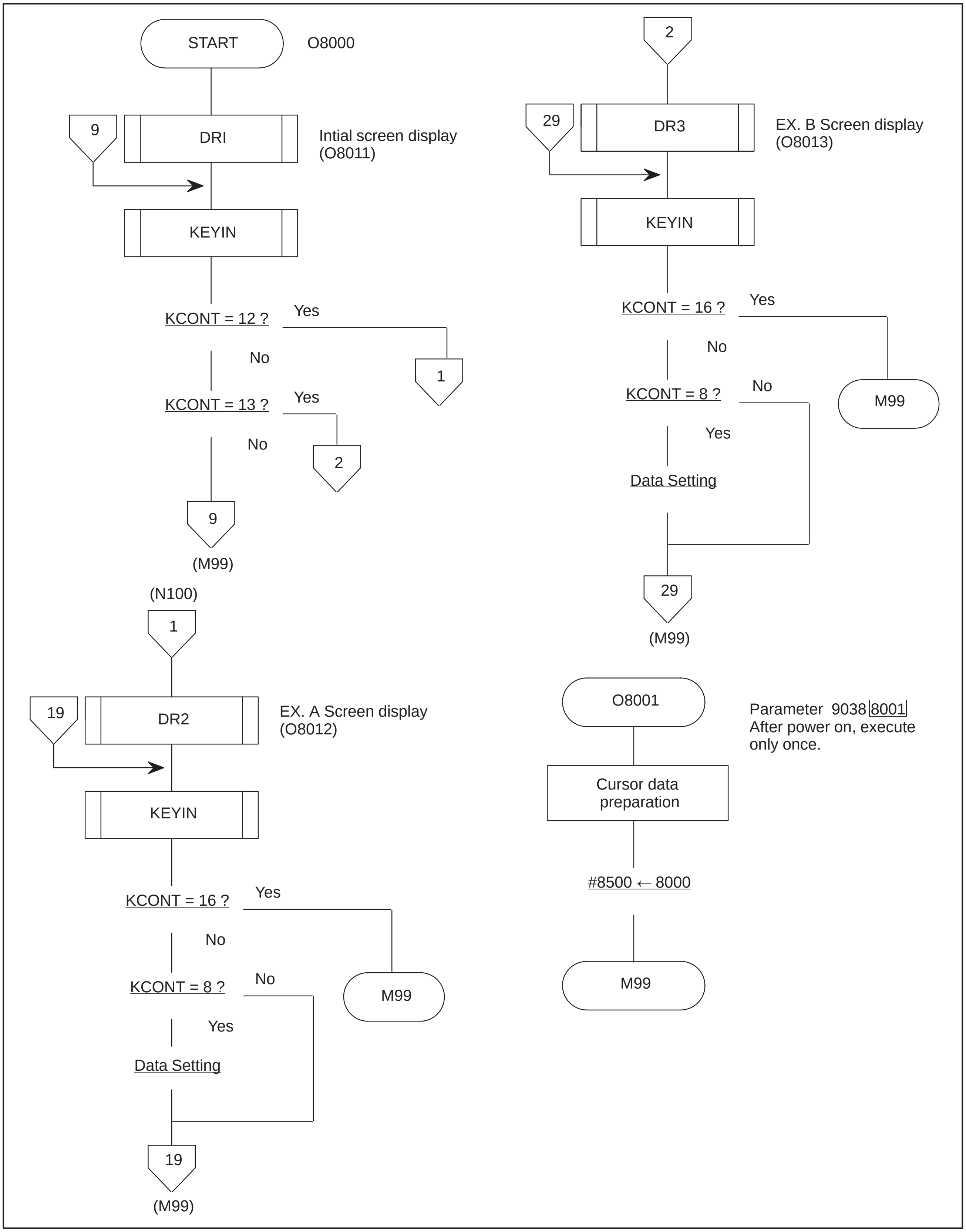
(3) Variables used in menu




## A.2.3 Program Structure



## A.2.4 Flow Chart



## A.2.5 Coding

(1) Main program

```
O8000 ;
N1 M98 P8011 ;                          Initial screen display
N9 M98 P9503 ;                          KEY IN
IF [#102 EQ 12] GOTO 100 ;              Soft key #1, #2 ?
IF [#102 EQ 13] GOTO 200 ;
M99 P9 ;
N100 M98 P8012 ;                        Screen (EX. A) display
N19 M98 P9503 ;
IF [#102 EQ 16] GOTO 199 ;              Soft key #5 (END) ?
IF [#102 NE 8] GOTO 190 ;               Input key judgment
IF [#104 EQ 1] GOTO 110 ;               Cursor position ?
#10050 = #103 ;
#140 = 4.0 ;
#141 = 0 ;                              PROG NO. setting
M98 P9501 ;
M99 P19 ;                               Processing on EX. A screen
N110 #[10051 + #105] = #103 ;
#140 =5.2 ;
#141 = 1 ;                              Data setting
M98 P9501 ;
N190 M99 P19 ;
N199 M99 P1 ;
N200 M98 P8013 ;                        Screen (EX. B) display
N29 M98 P9503 ;                         Key IN
IF [#102 EQ 16] GOTO 299 ;              Soft key (END) ?
IF [#102 NE 8] GOTO 290 ;               Input key judgment
IF [#104 EQ 1] GOTO 210 ;               Cursor position ?
#10060 = #103 ;
#140 = 4.0 ;
#141 = 0 ;                              PROG. NO. setting
M98 P9501 ;
M99 P29 ;                               Processing on FX. B screen
N210 #[10061 + #105] = #103 ;
#140 = 4.0 ;
#141 = 0 ;                              Data setting
M98 P9501 ;
N290 M99 P29 ;
N299 M99 P1 ;
;
;
;
;
;
```

(2) Initial screen display (Subprogram)

```
O8011 ;
M98 P9506 ;
G243 X0 Y1 A3 (MACRO EXAMPLE) ;
X2 Y4 A1 (SOFTKEY SELECT) ;
X3 Y6 (1 - EXAMPLE A) ;
X3 Y7 (2 - EXAMPLE B) ;
X3 Y8 (3 - EXAMPLE C) ;
X3 Y9 (4 - EXAMPLE D) ;
X3 Y10(5 - EXAMPLE E) ;
X2 Y16(EX.A) ;
X10(EX.B) ;
X18(EX.C) ;
X26(EX.D) ;
X34(EX.E) ;
#100= 0 ;
M99 ;
;
;
;
;
;
```

(3) EX. A Screen display (Subprogram)

```
O8012 ;
M98 P9506 ;
G243 X1 Y0 A1 (EXAMPLE A) ;
     X3 Y2    (PROG NO ) C61 ;
X0 Y4 (  DATA 1 ) C61 ;
X0 Y5 (  DATA 2 ) C61 ;
X0 Y6 (  DATA 3 ) C61 ;
X0 Y7 (  DATA 4 ) C61 ;
X0 Y8 (  DATA 5 ) C61 ;
X0 Y9 (  DATA 6 ) C61 ;
X0 Y10(  DATA 7 ) C61 ;
X34 Y16 (END) ;
G243 X14 Y2 F4.0 Z0 D#10050 ;
#140 = 5.2 ;
#141 = 1 ;
#142 = 11 ;
#143 = 4 ;
#144 = 10051 ;
```

```
#145 = 7 ;
#146 = -9999 ;
#147 = 1 ;
M98 P9505 ;
#100 = 1 ;
#104 = 0 ;
#105 = 0 ;
#140 = 0 ;
M98 P9502 ;
M99 ;
;
;
;
;
;
```

(4) FX.B Screen program (Subprogram)

```
O8013 ;
M98 P9506 ;
G243 X1 Y0 A1 (EXAMPLE B) ;
X3 Y2 (PROG NO)(*3D*) ;
X0 Y4 (DATA SET1) ;
X0 Y5 (DATA SET2) ;
X0 Y6 (DATA SET3) ;
X0 Y7 (DATA SET4) ;
X0 Y8 (DATA SET5) ;
X34 Y16 (END) ;
G243 X14 Y2 F4.0 Z0 D#10060 ;
#110 = 0 ;
WHILE [#110 LE 4] DO 1 ;
#140 = 4.0 ;
#141 = 0 ;
#142 = 12 ;
#143 = #110 + 4 ;
#144 = 10061 + #110 * 4 ;
#145 = 4 ;
#146 = -9999 ;
#147 = 6 ;
#98 P9504  ;
#110 = #110 + 1 ;
END 1 ;
#100 = 2 ;
```

```
#104 = 0 ;
#105 = 0 ;
#140 = 0 ;
M98 P9502 ;
M99 ;
;
;
;
;
;
;
```

(5) Cursor control data prepation

```
O8001 ;
#10000 = 10010 ;
#10001 = 2 ;
#10002 = 10026
#10003 = 2 ;
#10010 = 13 ;
#10011 = 2 ;
#10012 = 0 ;
#10013 = 0 ;
#10014 = 1 ;
#10015 = 1 ;
#10016 = 1 ;
#10017 = 0 ;
#10018 = 10 ;
#10019 = 4 ;
#10020 = 0 ;
#10021 = 1 ;
#10022 = 1 ;
#10023 = 7 ;
#10024 = 1 ;
#10025 = 1 ;
#10026 = 13 ;
#10027 = 2 ;
#10028 = 0 ;
#10029 = 0 ;
#10030 = 1 ;
#10031 = 1 ;
#10032 = 1 ;
#10033 = 0 ;
#10034 = 11 ;
```

```
#10035 = 4 ;
#10036 = 6 ;
#10037 = 1 ;
#10038 = 4 ;
#10039 = 5 ;
#10040 = 1 ;
#10041 = 0 ;
#8500 = 8000 ;
M99 ;
;
;
;
;
;
```

Make this program so that it is executed only once after power ON.

# A.3 STANDARD MACRO PROGRAM

## A.3.1 Standard Routine List

| No | Routine Name | Program number | Explanation |
|---|---|---|---|
| 1 | DDPL | O9501 | Data display to cursor position |
| 2 | CDPL | O9502 | Cursor display |
| 3 | KEYIN | O9503 | Key input control |
| 4 | VDPLX | O9504 | One- row display of variable data |
| 5 | VDPLY | O9505 | One- column display of variable data |
| 6 | DSPC | O9506 | Screen CLEAR |
| 7 | VSET | O9507 | Data setting to continuous variable area |
| 8 | VCOPY | O9508 | Variable copy |

## A.3.2 Area of Variable Used

In standard routine, the following variable area is used.

In user's program, do not use this area for others purposes.

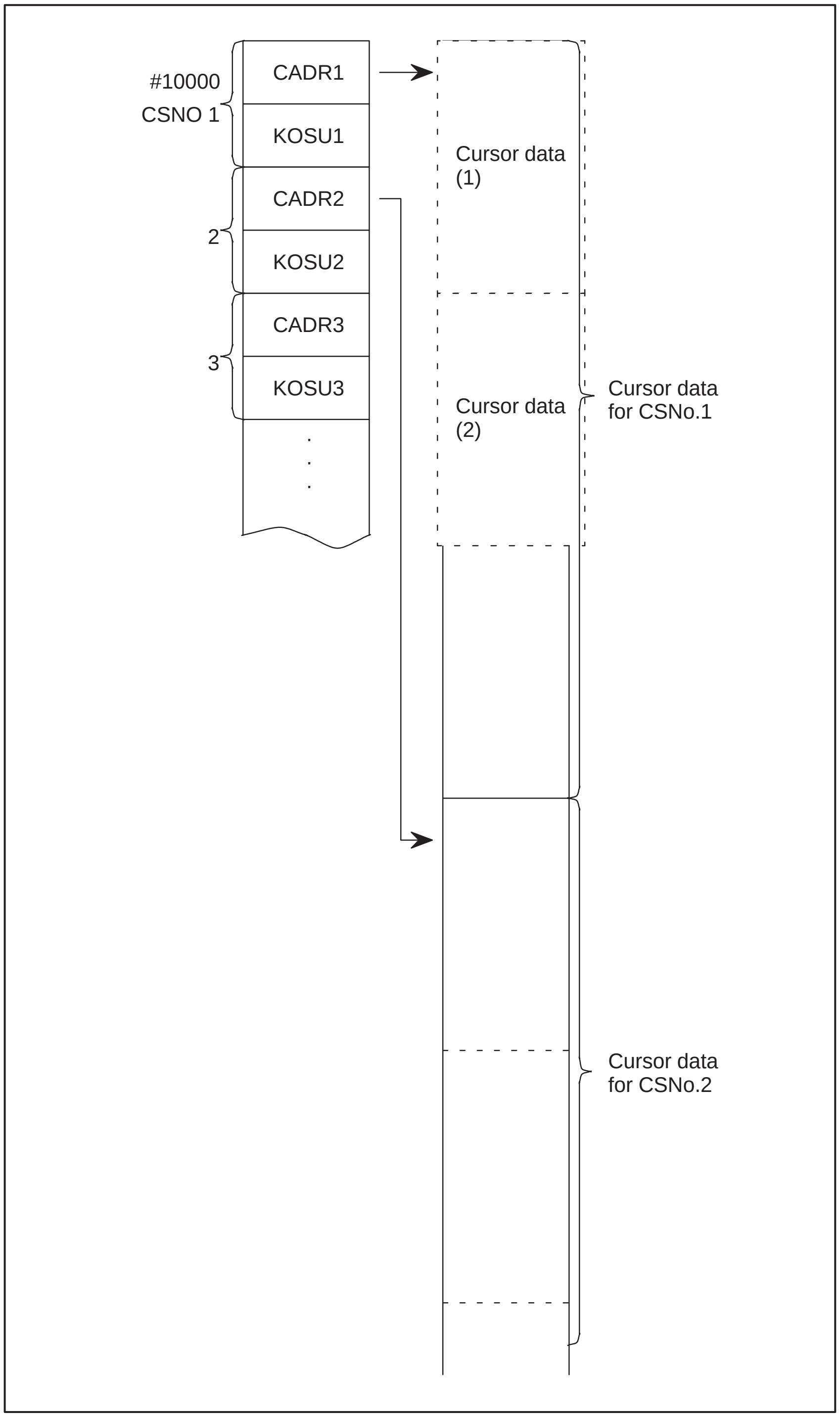
| | |
|---|---|
| P-CODE variables | #100 - #109<br>#139 - #148 |
| Common variables | #10000 - (up to the variables required) |

## A.3.3 Explanation of Variable Area

(1) P-CODE variable

| | | |
|---|---|---|
| #100 | CSNO | Cursor control No. of screen being now selected |
| 1 | Vacant | |
| 2 | KCONT | Key input control data (#8501 value) |
| 3 | KDATA | Key input numerical data (#8503 value) |
| 4 | CPNT | Cursor pointer (0,1, 2, .......) |
| 5 | SUBP | Cursor subpointer (0,1, 2, .......) |
| | Spare | |
| 9 | Spare | |
| #139 ~ #148 | Work area | |
| #149 | | For macro call using T code (RESERVE) |

(2) Cursor control data




1. CSNO (Cursor control No.)
   1, 2, 3...are assigned to screen No. requiring cursor control.
   In a program, to control the cursor on a screen, set this number to #100 value.
   When #100 is ”0”, the cursor is not displayed.
2. CADDRI
   It is necessary to prepare cursor data corresponding to cursor control No., but set its top address to CADDRI.
3. KOSUI
   Specify the number of cursor data required on one screen.

(3) Explanation of cursor data

| | | |
|---|---|---|
| CADRI | CPX | Cursor position X |
| | CPY | Cursor position Y |
| | C∆X | X increment for group: 0 for no group |
| | C∆Y | Y increment for group: 0 for no group |
| | CNX | X number for group: 1 for no group |
| | CNY | Y number for group:1 for no group |
| | CINP | Data input relative position |
| | CXYF | Cursor moving direction for group<br>0: Move in X direction<br>1: Move in Y direction |

1. Cursor data
   One cursor data corresponds to one cursor.
   However, when the cursors are arranged in line regularly, those cursors are regarded as one group, which corresponds to one cursor data.
2. Cursor position (X, Y)
   Specify the cursor position, using the coordinate on the screen.
   When the cursors form a group, specify the position of a cursor at the head of the group.
3. X increment and Y increment (∆X, ∆Y)
   When the cursors form a group, specify the spacing between the cursors in line.

   When the cursors are arranged only in the X-direction, specify 0 for ∆Y, and when the cursors are arranged only in the Y-direction, specify 0 for ∆X.
4. X number and Y number
   When the cursors form a group, specify the number of cursors in X-direction and in Y-direction.
   Specify 1 for Y and X number in case of X-direction only and Y-direction only arrangements respectively.
   When the cursors does not form a group, specify 1 for the both.
5. Data input relative position (CINP)
   When the key input numerical value is displayed at the cursor, this function displays it at the position deviated in X-direction by the specified figure.

   9999.999

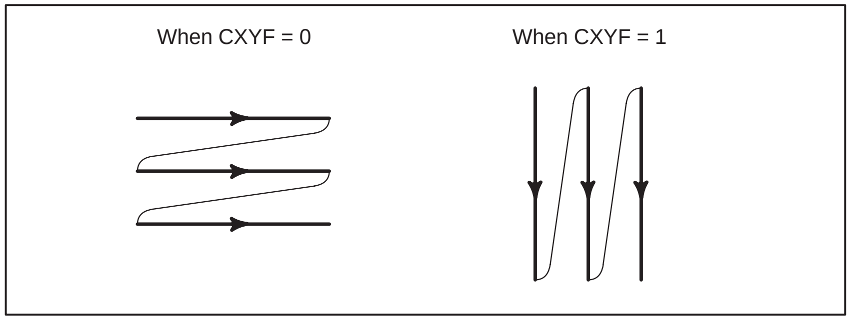
   _

   Cursor

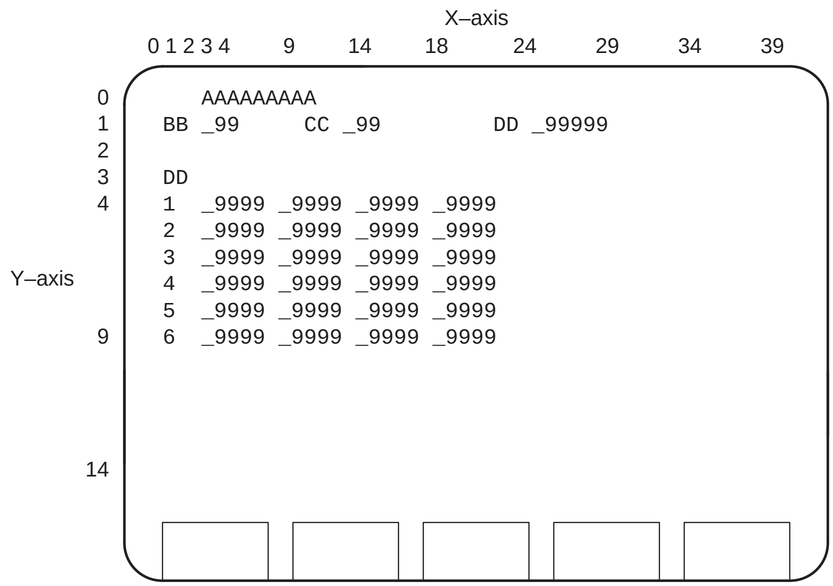
   In this case, ”5” should be set for CINT.

6. CXYF

Specify the direction of cursor movement within the group.



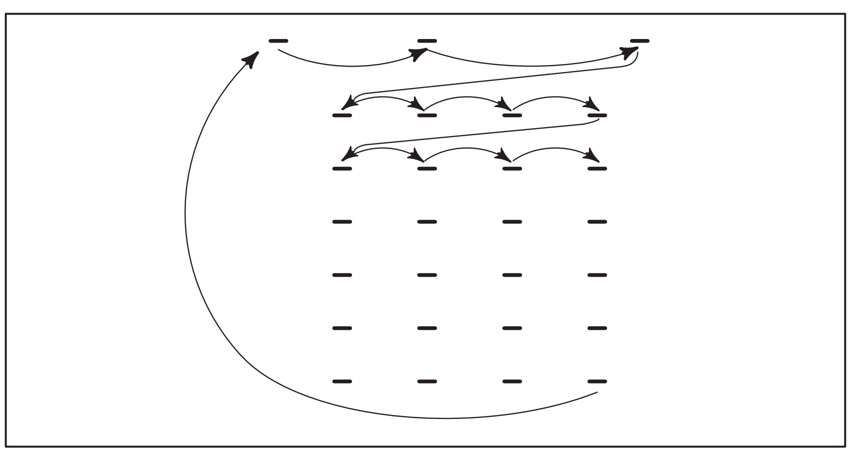

(4) Sample preparation of cursor data




The position of "_" in the following figure indicates the position where the cursor moves.

(Actually, the cursor is always displayed at one place.)

Cursor movement



(5) Cursor data

| | |
|---|---|
| #10000 | 10010 |
| 1 | 3 |

| | | |
|---|---|---|
| #10010 | 4 | BB/CC CURSOR |
| 1 | 1 | |
| 2 | 9 | |
| 3 | 0 | |
| 4 | 2 | |
| 5 | 1 | |
| 6 | 1 | |
| 7 | 0 | |
| 8 | 28 | D cursor |
| 9 | 1 | |
| 20 | 0 | |
| 1 | 0 | |
| 2 | 1 | |
| 3 | 1 | |
| 4 | 1 | |
| 5 | 0 | |
| 6 | 7 | Matrix cursor |
| 7 | 4 | |
| 8 | 7 | |
| 9 | 1 | |
| 30 | 4 | |
| 1 | 6 | |
| 2 | 1 | |
| 3 | 0 | |

## A.3.4 Standard Routine

To use the routines (O9501, O9502, O9503) with which the cursor position is associated among the following routines, the cursor data corresponding to the screen selected at that time should be selected. Specifically, set the following;

P-CODE variables

#100 = Cursor control No. of screen being selected at present

#104 = Cursor pointer (Initial value 0)

#105 = Cursor subpointer (Initial value 0)

#104 and #105 values are automatically changed each time the cursor key is pressed.

On the screen with no cursor, set #100 to ”0”

(1) DDPL O9501

1. Function

   This function displays the input data variable (KDATA value) in the position deviated in the X-direction by ”CINT” from the cursor position.

2. Calling format

   #140= f (F) : Total digits and digits under decimal point

   #141= z (Z) : Reading zero

   M98 P9501

   f and z are the values commanded by G243.

3. Remark

   When this routine is executed , the cursor display position advances by one.

   #140 and #141 values are saved.

(2) CDPL O9502

1. Function

   This function displays the cursor.

   The cursor position is determined by #100 - #105 and #10000- values.

2. Calling format

   #140=n

   M98 P9502

   n= 0 : No cursor pointer changes.

   1 : The cursor pointer is advanced by one, displaying the cursor.

   2 : The cursor pointer is retracted by one, displaying the cursor.

3. Remark

   After this routine is executed, the #140 value is saved.

(3) KEYIN O9503

1. Function

   This function accepts key input of numerical data, executing the following processing with its value.

   (a) Cursor key

   ↓ : Moves the cursor forward by one.

   ↑ : Moves the cursor backward by one.

   (b) Other than the cursor key

   #8501 and #8503 values are saved.

   #102(KCONT) ←#8501

   #103(KDATA) ←#8503

2. Calling format

   M98 P9503

(4) VDPLX O9504

1. Function

   To display a continuous variable value, this function deviates it in the X-direction from the specified place on the CRT screen. (Within one line)

2. Calling format

   #140=F (F)

   #141=Z (Z)

   #142=X Display position on CRT

   #143=Y

   #144=V Variable No.(head)

   #145=N No. of display

   #146=E End code

   #147=∆X Deviation in the X-direction on the CRT

   M98 P9504

   E : END code

   The display ends when the data which coincides with the end code is reached even within the number of displays.

3. Remark

   After this routine is executed, #140 - #147 values are saved.

(5) VDPLY O9505

1. Function

   To display a continuous variable value, this function deviates it in the Y-direction from the specified place on the CRT screen. (within one column)

2. Calling format

   #140=F

   #141=Z

   #142=X Display position on CRT

   #143=Y

   #144=V Variables No. (head)

   #145=N No. of displays

   #146=E End code

   #147=∆Y Deviation in Y-direction on the CRT

   M98 P9505

   E : End code

   The display ends when the data which coincides with the end code is reached even within the number of displays.

3. Remark
   After this routine is executed, #140 - #147 values are saved.

(6) DSPC O9506

1. Function
   This function clears the screen, including the soft key.

2. Calling format
   M98 P9506

3. Remark
   The soft key is not cleared with G202.

(7) VSET O9507

1. Function
   This function sets the specified data to a continuous variable area.
2. Calling format

   #140 = V Variable No. (head number)
   #141 = D Data to be set
   #142 = N No. of variables

   M98 P9507
3. Remark
   After this routine is executed, #140 - #142 are saved.

(8) VCOPY O9508

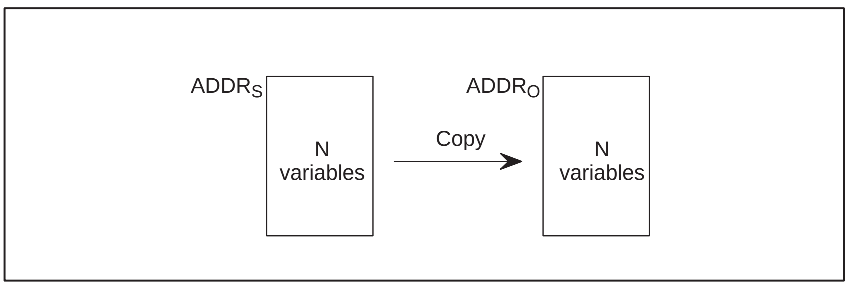
1. Function
   This function copies the data in a continuous variable area to another area.
2. Calling format

   #140=$ADDR_S$ Address of original data to be copied

   #141=$ADDR_O$ Address of data to be copied

   #142=N No. of variables

   M98 P9508




3. Remark
   After this routine is executed, the #140 - #142 values are saved.

## A.3.5 List for Source Program

```
O9501 ;
IF [#100 EQ 0] GOTO 900 ;
#145 = #[10000 + [#100 - 1] * 2] ;
IF [#145 EQ 0] GOTO 900 ;
#145 = #104 * 8 + #145 ;
IF [#[#145+7] EQ 1] GOTO 10 ;
#139 = FIX [#105 / #[#145 + 4]] ;
#148 = #105 - #139 * #[#145 + 4] ;
GOTO 20 ;
N10 #148 = FIX  [#105 / #[#145 + 5]] ;
#139 = #105 - #148 * #[#145 + 5] ;
N20 G243 F#140 Z#141 ;
X[#[#145]+#[#145+2]*#148+#[#145+6]] Y[#[#145+1]+#[#145+3]*#139]
D#103
#147 = #140 ;
#140 = 1 ;
M98 P9502 ;
#140 = #147 ;
N900 M99 ;
  ;
  ;
  ;
  ;
  ;
```

```
O9502 ;
IF [#100 EQ 0] GOTO 900 ;
#145 = #[10000 + [#100-1] * 2] ;
IF [#145 EQ 0] GOTO 900 ;
#145 = #104 * 8 + #145 ;
#146 = #[#145 + 4] * #[#145+5] ;
#147 =#[10001 + [#100-1] * 2] ;
IF [#140 EQ 0] GOTO 200 ;
IF [#140 EQ 2] GOTO 100 ;
#105 = #105 + 1 ;
```

```
IF [#105 LT #146] GOTO 200 ;
#105 = 0 ;
#104 = #104 +1 ;
#145 = #145 + 8 ;
IF [#104 LT #147] GOTO 200 ;
#104 = 0 ;
#145 = #[10000 + [#100 -1] * 2] + #104 * 8 ;
GOTO 200 ;
N100 #105 = #105 - 1 ;
IF [#105 GE 0] GOTO 200 ;
#104 = #104 - 1 ;
IF [#104 GE 0] GOTO 120 ;
#104 = #147 - 1 ;
N120 #145 = #[10000 + [#100 -1] * 2] + #104 * 8 ;
#105 = #[#145 + 4] * #[#145 + 5] - 1 ;
N200 IF [#[#145 + 7] EQ 1] GOTO 210 ;
#139 = FIX [#105 / #[#145 + 4]] ;
#148 = #105 - #139 * #[#145 + 4] ;
GOTO 220 ;
N210 #148 = FIX [#105 / #[#145 + 5]] ;
#139 = #105 - #148 * #[#145 + 5] ;
N220 #8505 = 1 ;
#8506 = #[#145] + #[#145 + 2] * #148 ;
#8507 = #[#145 + 1] + #[#145 + 3] * #139 ;
N900 M99 ;
  ;
  ;
  ;
  ;
  ;
```

```
O9503 ;
#8502 = 1 ;
#102 = #8501 ;
#103 = #8503 ;
#140 = 1 ;
IF [#102 EQ 3] GOTO 10 ;
IF [#102 NE 4] GOTO 20 ;
#140 = 2 ;
N10 M98 P9502 ;
N20 M99 ;
  ;
```

```
 ;
 ;
 ;
 ;
O9504 ;
G243 F#140 Z#141 ;
#139 = 0 ;
WHILE [#139 LT #145] DO 1 ;
IF [#[#144 + #139] EQ #146] GOTO 90 ;
X[#142 + #147 * #139] Y#143 D#[#144 + #139]
#139 = #139 + 1 ;
END 1 ;
N90 M99 ;
 ;
 ;
 ;
 ;
 ;
```

```
O9505 ;
G243 F#140 Z#141 ;
#139 = 0 ;
WHILE [#139 LT #145] DO 1 ;
IF [#[#144 + #139] EQ #146] GOTO 90 ;
X#142 Y[#143 + #147 * #139] D#[#144 + #139] ;
#139 = #139 + 1 ;
END 1 ;
N90 M99 ;
 ;
 ;
 ;
 ;
 :
```

```
09506 ;
G243 X0 Y0 K560 ;
#8505 = 0 ;
X1 Y16 K6 ;
X9 K6 ;
X17 K6 ;
X25 K6 ;
X33 K6 ;
```

```
M99 ;
 ;
 ;
 ;
 ;
 ;
```

```
O9507 ;
#139 = 0 ;
WHILE [#139 LT #142] DO 1 ;
#[#140 + #139] = #141 ;
#139 = #139 + 1 ;
END 1 ;
M99 ;
 ;
 ;
 ;
 ;
 ;
```

```
O9508 ;
#139 = 0 ;
WHILE [#139 LT #142] DO 1 ;
#[#141 + #139] = #[#140 + #139] ;
#139 = #139 + 1 ;
END 1 ;
M99 ;
```

## A.3.6 Explanation of Program

```
O9501 ; Display of data in cursor position
IF [#100 EQ 0] GOTO 900 ; #100: Cursor control No.
#145 = #[10000 + [#100 – 1] * 2] ; #145: Cursor data head address
IF [#145 EQ 0] GOTO 900 ;
#145 = #104 * 8 + #145 ; #104: Cursor pointer/display cursor address calculation
IF [#[#145+7] EQ 1] GOTO 10 ; Display direction?
#139 = FIX [#105 / #[#145 + 4]] ; When displaying the data in the X-direction
#148 = #105 - #139 * #[#145 + 4] ;
GOTO 20 ;
N10 #148 = FIX [#105 / #[#145 + 5]] ; When displaying the data in the Y-direction
#139 = #105 - #148 * #[#145 + 5] ;
N20 G243 F#140 Z#141 ; Format setting for data display
X[#[#145]+#[#145+2]*#148+#[#145+6]] Y[#[#145+1]+#[#145+3]*#139]
D#103
#147 = #140 ;         Stack #140. #140=1
#140 = 1 ;
M98 P9502 ; Cursor display subrutine
#140 = #147 ; Return #140 to this intial state.
N900 M99 ;
  ;
  ;
  ;
  ;
  ;
```

```
O9502 ; Cursor display
IF [#100 EQ 0] GOTO 900 ;
#145 = #[10000 + [#100-1] * 2] ;
IF [#145 EQ 0] GOTO 900 ; #145: Cursor data head address
#145 = #104 * 8 + #145 ;
#146 = #[#145 + 4] * #[#145+5] ; No. of displays in group
#147 =#[10001 + [#100-1] * 2] ; No. of cursor pointers
IF [#140 EQ 0] GOTO 200 ; No cursor pointer changes
IF [#140 EQ 2] GOTO 100 ;
#105 = #105 + 1 ;          ↓ cursor pointer advances
IF [#105 LT #146] GOTO 200 ;
#105 = 0 ; #105 is pointer No. in group
#104 = #104 +1 ;                                    When in the last poinnter is reached
#145 = #145 + 8 ;          Next group
IF [#104 LT #147] GOTO 200 ;     Return to the begining when coming to the
#104 = 0 ;                       cursor pointer group end
#145 = #[10000 + [#100 -1] * 2] + #104 * 8 ;
GOTO 200 ;
N100 #105 = #105 - 1 ;        ↓Cursor pointer moves backward
IF [#105 GE 0] GOTO 200 ;
#104 = #104 - 1 ;
IF [#104 GE 0] GOTO 120 ;
#104 = #147 - 1 ;
N120 #145 = #[10000 + [#100 -1] * 2] + #104 * 8 ;
#105 = #[#145 + 4] * #[#145 + 5] - 1 ;
N200 IF [#[#145 + 7] EQ 1] GOTO 210 ; ↓ When the cursor does not change.
#139 = FIX [#105 / #[#145 + 4]] ;      The cursor pointer moves in the X-direction
#148 = #105 - #139 * #[#145 + 4] ;     in a gruop: #148: X #149: Y
GOTO 220 ;
N210 #148 = FIX [#105 / #[#145 + 5]] ;     The cursor pointer moves in the Y-
#139 = #105 - #148 * #[#145 + 5] ;         direction in a group.
N220 #8505 = 1 ; Cursor display ON
#8506 = #[#145] + #[#145 + 2] * #148 ; Cursor X position
#8507 = #[#145 + 1] + #[#145 + 3] * #139 ; Cursor Y position
N900 M99 ;
; #140=0 : No cursor pointer changes                 #100  Cursor control1 No.
; #140=1 : The cursor pointer moves forward by one.   #101
; #140=2 : The cursor pointer moves backward by one.) #102  Key input control data
;                                                     #103  key input numerical data
;                                                     #104  Cursor pointer
                                                      #105  Cursor subpointer
```

```
O9503 ; Key input control
#8502 = 1 ; Numerical data input
#102 = #8501 ; Key input variable read
#103 = #8503 ; Numerical data read
#140 = 1 ; Advance the cursor.
IF [#102 EQ 3] GOTO 10 ; 3:For cursor↓, advance the cursor pointer.
IF [#102 NE 4] GOTO 20 ; 4:For cursor↑, reaturn the cursor pointer.
#140 = 2 ; Cursor moves backward
N10 M98 P9502 ; Cursor diaplsy sub
N20 M99 ;
 ;
 ;
 ;
 ;
 ;
```

```
O9504 ; One-line display of variable data (X axis direction)
G243 F#140 Z#141 ; Display format setting
#139 = 0 ; Counter
WHILE [#139 LT #145] DO 1 ; #145 in No. of displays.
IF [#[#144 + #139] EQ #146] GOTO 90 ;
X[#142 + #147 * #139] Y#143 D#[#144 + #139] ;
#139 = #139 + 1 ; #144 Head of variable No.
END 1 ;            #146  END code → If the display data is the same as
                          END code, the cursor RETURNS with no display.
N90 M99 ;
;
;                  #147 Deviation in X direction
;
;
;
```

```
O9505 ; One-row display of variable data (Y direction)  Difference from O09504
                                                        is that #147 is the
G243 F#140 Z#141 ; Display format setting               deviation in Y direction
#139 = 0 ;Counter
WHILE [#139 LT #145] DO 1 ;
IF [#[#144 +#139] EQ #146] GOTO 90 ;
X#142 Y[#143 + #147 * #139] D#[#144 + #139] ;
#139 = #139 +1 ;
END 1 ;
N90 M99 ;
 ;
 ; ’
```

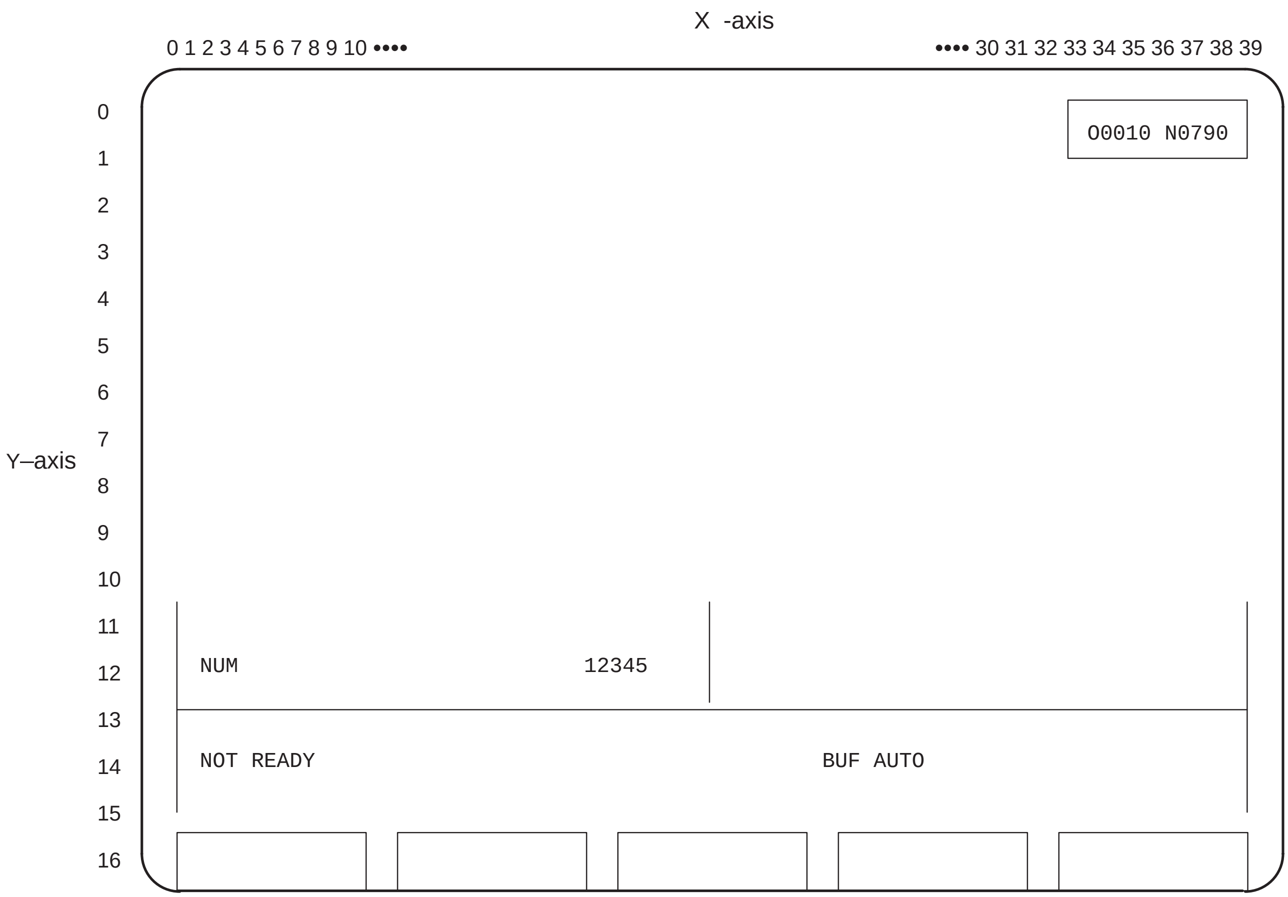
```
 ;
 ;
 ;
```

```
O9506 ; Screen clear
G243 X0 Y0 K560 ; Blank of 560 value
#8505 = 0 ; Cursor off
X1 Y16 K6 ;
X9 K6 ;                                      Leaving the "frame" of the soft
X17 K6 ;          Erase the soft key          key, erase the other part.
X25 K6 ;
X33 K6 ;
M99 ;
 ;
 ;
 ;
 ;
 ;
```

```
O9507 ; Set data in continued data area.
#139 = 0 ;
WHILE [#139 LT #142] DO 1 ;
#[#140 + #139] = #141 ;
#139 = #139 + 1 ;
END 1 ;
M99 ;
 ;
 ;
 ;
 ;
 ;
```

```
O9508 ; Variable copy
#139 = 0 ;
WHILE [#139 LT #142] DO 1 ;
#[#141 + #139] = #[#140 + #139] ;
#139 = #139 + 1 ;
END 1 ;
M99 ;
```

# A.4 SCREEN LAYOUT



# B MACRO COMPLIER OPERATION (Series 16–A/18–A)

**NOTE**

For operation of macro compiler of Series 20/21, refer to FAPT MACRO COMPILER (For personal computer) Programming Manual (B-66102E).

# B.1 OPERATION ON P–G

## B.1.1 Special Notes

First, special notes on operation are described below.

(1) When coming to a deadlock, press **<NL>** key.

Press **<NL>** key several times. The display returns to the initial screen.

Next, checking what should be operated, proceed with operation.

When ”FD0 =⌴ ” is displayed at the lower left part of the screen, **<F2>** or **<F7>** may be ON.

At that time, to return the display to the initial screen, turn **<F2>** or **<F7>** OFF, and key in NO **<NL>**.

(2) Remember to switch!

Do not leave F key (**<F0> - <F7>**) ON. Before pressing R key, be sure to check F key ON/OFF.

(3) Numeric 1 and alphabet I

Numeric 0 and alphabets O, Comma, and point.

Most of input errors result from these figures, characters and symbols.

(4) Do not forget the minus (–) symbol.

## B.1.2 Basic Operations

### B.1.2.1 Connectiing to FA writer

Connect FA writer to CN2 or CN3 of P-G. P-G operation is as follows.
(Return to) Initial Screen and press ”R1”

Answer the inquiry on the screen with NO = 2 **<NL>**

## B.1.3 Inputting Macro Program from Keyboard

(Return to) Initial Screen, **<F0> - <F7>** OFF and press ”R2”

NO = 1 **<NL>**

PROGRAM = IN **<NL>**

INPUT = O Program No. **<NL>** (First register Program No. only)

INPUT = **<NL>** only

PROGRAM = O Program No. **<NL>** (Previously input number)

EDIT = K⌴ OLD **<NL>** (⌴ Space key)

Line No. at the lower left part of the screen is from O0020 on.

(1) Input each line **<NL>**.

(2) Before keying in **<NL>**, if a key-in error is found, use BS/CAN keys to correct it.

(3) Note that program input ends for **<NL>** only.

(4) After one line **<NL>**, if a key operation error is found, press the ”CHG” key. (Check that F15 key lights up.)

Skip the cursor at the left upper part of the screen and move it to the place to be corrected, using the CURSOR key.

1. ”CHG” screen edit switching (F15 key lights up)
2. ”INS” insertion: the line is spaced out by a push on this key when the cursor is located in Line No.
3. ”DEL” deletion: the line is deleted by a push on this key when the cursor is located in Line No.
4. ”R1” Correction/cancellation: the current data is corrected and canceled on the screen.

   The cursor is moved to the upper left part of the screen.
5. ”R2” Page shift backward , ”R3” Page shift forward

(5) After correction, press the ”CHG” key once again. (Check that F15 key goes off.)

The cursor returns to the lower left part of the screen. (The left lower screen indicates Line No.) : so input data successively.

(6) To space out a line, input space **<NL>**.

(7) After keying in the last line, this operation ends by a push on **<NL>** only. ”EDIT = ” is displayed at the left lower part of the screen.

(8) Further, when it is desired to input the next program, press **<NL>** once again:

”PROGRAM = ” is displayed at the left lower part of the screen. Operate this with IN **<NL>** .

(9) Press **<NL>** several times : the display returns to the initial screen.

## B.1.4 Correcting Macro Program (Screen edit)

(Return to) Initial Screen **<F0> - <F2>** OFF and press ”R2”

NO=1 **<NL>**

PROGRAM = O program No. **<NL>** (Number to be corrected)

EDIT=

(1) Screen Edit

- Press the ”CHG” key. (Check that F15 key lights up)

  Skip the cursor at the upper left part of the screen, and move it to the place to be corrected, using the CURSOR key. After completion of correction, return the ”CHG” key to the lower left part of the screen again.

  ”CHG”: Screen edit switching (F15 key ON)

  ”INS”: Insertion: the line is spaced out by a push on this key when the cursor is located in Line No.

  ”DEL”: Deletion: the line is deleted by a push on this key when the cursor is located in Line No. (Do this most carefully.)

  ”R1”: Correction/cancellation: the current data is corrected and canceled on that screen.

  The cursor is moved to the upper left part of the screen.

  ”R2”: Page shift backward

  ”R3”: Page shift forward

- After correction, press the ”CHG” key once again. (Check that F15 key goes off.)

  The cursor returns to the lower left part of the screen.

  (”EDIT = ” is displayed at the lower part left of the screen.)

## B.1.5 Correcting Macro Program (Command edit)

(Return to) Initial Screen **<F0>** - **<F2>** OFF and press ”R2”

NO=1 **<NL>**

PROGRAM =O Program No. **<NL>** (Number to be corrected)

EDIT=

Command Edit

- Check that left lower part of the screen is ”EDIT = ”.
- Check that the cursor is located at the left lower part of the screen.
  - Replacement : Line No.␣ Program one line **<NL>**
  - Insertion : New Line No.␣ Program one line **<NL>**
  - One-line deletion : Line No. **<NL>**
  - Plural deletion : Start Line No., End Line No. **<NL>**
  - Display : L␣ Line No. **<NL>**
  - Line renewal : REN **<NL>**
  - Copy : MOVE (Start Line No., End Line No. )<br>␣ /Frequency **<NL>**<br>MOVE␣ Frequency **<NL>**
  - Substitute : A (Start Line No., End Line No.) ␣<br>/Original character string/<br>New character string /All **<NL>**<br>A␣ /Original character string/,<br>/New character string/, /All **<NL>**
  - Character search : After **<NL>** only L**<NL>** and F/<br>Character string / **<NL>**

## B.1.6 Outputting Files to Floppy Disk

(Return to) Initial Screen **<F7>** ON : **<F5>** ON as needed and press ”R2”

NO = 1 **<NL>**

PROGRAM = OUT␣ ALL **<NL>** or OUT␣ O (Program No.) **<NL>**

FD0= (For first time)

$$\text{OK}\left\{\begin{matrix}\text{I7}\\ \text{INT}\end{matrix}\right\}\sqcup\ \text{, / Date ,@ File name } \textbf{(NL)} \left\{\begin{matrix}0\\ \hline 1\end{matrix}\right\}$$

FD0= (2nd time and after)

$$\text{OK}\sqcup\text{/ Date ,}\left\{\begin{matrix}0\\ \hline 1\end{matrix}\right\}\ \text{@ File name } \textbf{(NL)}$$

Upon completion, turn **<F7>**OFF, and **<F5>** OFF as needed.

$$\text{FD0= OK }\sqcup\left\{\begin{matrix}\text{ADD}\\ \text{INT}\\ \text{I2 – I17}\end{matrix}\right\}\underline{\text{, / Date}}\left\{\begin{matrix}\text{NP}\\ \hline \text{P}\end{matrix}\right\},\left\{\begin{matrix}0\\ \hline 1\end{matrix}\right\}\left\{\begin{matrix}\text{@ File name}\\ \text{: File No.}\end{matrix}\right\}\text{”\_” is omissible}$$

- 71 files can be registered by INT and 175 files can be registered by I2 for the first time. Consequently, 695 files can be registered for I7. (incremented by 104)
- In the file with same name, no substituting processing is carried out, so that no output is possible.
- The file name consists of up to 17 character, ”,” comma cannot be used.
- A longer file name is preferable for easier identification.·
- At input, File No. can be used for access.

## B.1.7 Inputting Files to P-G from Floppy Disk

(Return to) Initial Screen only **<F2>** ON and press ”R2”

NO = 1 **<NL>**

PROGRAM = IN **<NL>**

FD0 = OK␣ $\left\{\begin{matrix}0\\1\end{matrix}\right\}\left\{\begin{matrix}\text{@ File name}\\\text{: File No.}\end{matrix}\right\}$ **<NL>**

Upon completion, turn **<F2>** OFF.

## B.1.8 Handling Floppy Disk

(1) File name directory

(Return to) Initial Screen and turn **<F5>** ON as needed and press R3.

REQUEST = FD LIST ␣ L, $\left\{\begin{matrix}0\\1\end{matrix}\right\}$ **<NL>**

1 sector = 256 characters

Converted to No. of characters if the size is multiplied by 256.

Upon completion, turn **<F5>** OFF as needed. ”_” is omissible

(2) File delete

(Return to) Initial Screen and press ”R3”

REQUEST = SCRATCH ␣ $\left\{\begin{matrix}1\\0\end{matrix}\right\}\left\{\begin{matrix}\text{@ File name}\\\text{: File No.}\end{matrix}\right\}$ **<NL>**

FD0 = OK **<NL>** In error FD0 = NO **<NL>**

(3) File delete area open

(Return to) Initial Screen and press ”R3”

REQUEST = CONDENCE␣ $\left\{\begin{matrix}0\\1\end{matrix}\right\}$ **<NL>**

(4) File delete area open

(Return to) Initial Screen and press ”R3.

REQUEST = RENAME␣ $\left\{\begin{matrix}0\\1\end{matrix}\right\}\left\{\begin{matrix}\text{@ File name}\\\text{: File No.}\end{matrix}\right\}, \left\{\begin{matrix}\text{NP}\\\text{P}\end{matrix}\right\}$ , /Date , @ New File name **<NL>**

(5) File attribute change

(Return to) Initial Screen and press ”R3”

<u>REQUEST</u> = REMOVE **<NL>**

FD0 = OK␣ $\left\{\begin{array}{c} \text{M} \\ \hline \text{A} \end{array}\right\}$ , Input driveNo. $\left\{\begin{array}{c} 0 \\ \hline 1 \end{array}\right\}\left\{\begin{array}{l} \text{@ File name} \\ \text{: File No.} \end{array}\right\}$ **<NL>**

”A : automatically”

”M : Ask for each time”

FD1 = OK␣ $\left\{\begin{array}{l} \text{I7} \\ \text{INT} \end{array}\right\}$ ,output drive No. $\left\{\begin{array}{c} 1 \\ \hline 0 \end{array}\right\}$ **<NL> (For the first time)**

FD1 = OK␣ Output drive No. $\left\{\begin{array}{c} 1 \\ \hline 0 \end{array}\right\}$ **<NL>**

# C DATA ON MACRO COMPILER

## C.1 ARGUMENT TRANSFER

It is possible to specify parameters for macro calls that can be referred to as local variables by the P-CODE program. In the parameter specification method for the custom macro of the custom macro B type, the parameter specification I and parameter specification II can be specified separately or mixed in one label.

Signs, decimal points, and other symbols not related to the address can be used in the parameters.

**Table C.1 (a) Argument specification I for P-CODE program call**

| Address of argument specification I | Local variable No. |
|---|---|
| A | #1 |
| B | #2 |
| C | #3 |
| I | #4 |
| J | #5 |
| K | #6 |
| D | #7 |
| E | #8 |
| F | #9 |
| G (Note) | #10 |
| H | #11 |
| L (Note) | #12 |
| M | #13 |
| N (Note) | #14 |
| P (Note) | #16 |
| Q | #17 |
| R | #18 |
| S | #19 |
| T | #20 |
| U | #21 |
| V | #22 |
| W | #23 |
| X | #24 |
| Y | #25 |
| Z | #26 |

**NOTE**

It is available when compilation parameter bit 5 of No.9008 is 1.

**Table C.1 (b) Argument specification II for P-CODE program call**

| Address of argument specification II | Local variable No. |
| --- | --- |
| A | # 1 |
| B | # 2 |
| C | # 3 |
| I1 | # 4 |
| J1 | # 5 |
| K1 | # 6 |
| : | : |
| I10 | # 31 |
| J10 | # 32 |
| K10 | # 33 |

## C.2 SUMMARY OF SYSTEM VARIABLES (Refer to Operator’s Manual for details)

## (1) Series 16-MB/16-MA/18-MB/18-MA/20-FA/21-MB

Tool compensation

| | |
|---|---|
| #2001 | Tool compensation value 1 |
| #2002 | ″ ″ 2 |
| ≀ | ≀ |
| #2200 | ″ ″ 200 |

Work offset

| | |
|---|---|
| #2500 | X. External work OFFSET |
| #2501 | G54 ″ ″ |
| ≀ | ≀ |
| #2506 | G59 ″ ″ |
| #2600 | Y. External work OFFSET |
| #2601 | G54 ″ ″ |
| ≀ | ≀ |
| #2606 | G59 ″ ″ |
| #2700 | Z. External work OFFSET |
| #2701 | G54 ″ ″ |
| ≀ | ≀ |
| #2706 | G59 ″ ″ |
| #2800 | 4. External work OFFSET |
| #2801 | G54 ″ ″ |
| ≀ | ≀ |
| #2806 | G59 ″ ″ |

G code

| | |
|---|---|
| #4001 | G00, 01, 02, 03, 33 |
| #4002 | G17, 18, 19 |
| #4003 | G90, 91 |
| #4004 | * |
| #4005 | G94, 95 |
| #4006 | G20, 21 |
| #4007 | G40, 41, 42 |
| #4008 | G43, 44, 49 |
| #4009 | G73, 74, 76, 80–89 |
| #4010 | G98, 99 |
| #4011 | G50, 51 |
| #4012 | G65, 66, 67 |
| #4013 | * |
| #4014 | G54–59 |
| #4015 | G61–64 |
| #4016 | G68 ,69 |
| #4017 | * |
| ≀ | ≀ |
| #4022 | * |

* :Reserved

Code

| | |
|---|---|
| #4102 | B code |
| #4109 | F code |
| #4111 | H code |
| #4113 | M code |
| #4114 | Sequence Number |
| #4115 | Program Number |
| #4119 | S code |
| #4120 | T code |

Position data

| | |
|---|---|
| #5001 | X. Block end position |
| #5002 | Y. ″ |
| #5003 | Z. ″ |
| #5004 | 4. ″ |
| #5021 | X. Machine coordinates |
| #5022 | Y. ″ |
| #5023 | Z. ″ |
| #5024 | 4. ″ |
| #5025 | 5. ″ |
| #5026 | 6. ″ |
| #5041 | X. Work coordinates |
| #5042 | Y. ″ |
| #5043 | Z. ″ |
| #5044 | 4. ″ |
| #5045 | 5. ″ |
| #5046 | 6. ″ |
| #5061 | X. Skip signal position |
| #5062 | Y. ″ |
| #5063 | Z. ″ |
| #5064 | 4. ″ |
| #5081 | X. Tool compensation value |
| #5082 | Y. ″ |
| #5083 | Z. ″ |
| #5101 | X. Servo variation value |
| #5102 | Y. ″ |
| #5103 | Z. ″ |
| #5104 | 4. ″ |

Control variables

| | |
|---|---|
| #3000 | Macro ALARM n<99 |
| #3001 | Clock (msec) |
| #3002 | Clock (Hour) |
| #3003 | Single block |
| #3004 | Feed hold |
| #3005 | Setting |
| #3011 | Clock (mm dd yy) |
| #3012 | Clock (hour minute second) |
| #3901 | Number of processed parts |
| #3902 | Number of required parts |

#3003

| n | SBK | FIN |
|---|---|---|
| 0 | ○ | w |
| 1 | × | w |
| 2 | ○ | × |
| 3 | × | × |

○ : No control
× : Control
W : Waiting for auxiliary function end signal

#3004

| n | FH | OR | ES |
|---|---|---|---|
| 0 | ○ | ○ | ○ |
| 1 | × | ○ | ○ |
| 2 | ○ | × | ○ |
| 3 | × | × | ○ |
| 4 | ○ | ○ | × |
| 5 | × | ○ | × |
| 6 | ○ | × | × |
| 7 | × | × | × |

FH : Feed hold
OR : Overwrite
ES : Exact stop check

Parameter instructions

| | | |
|---|---|---|
| | | #0 |
| A | A | #1 |
| B | B | #2 |
| C | C | #3 |
| I | I1 | #4 |
| J | J1 | #5 |
| K | K1 | #6 |
| D | I2 | #7 |
| E | J2 | #8 |
| F | K2 | #9 |
| | I3 | #10 |
| H | J3 | #11 |
| | K3 | #12 |
| M | I4 | #13 |
| | J4 | #14 |
| | K4 | #15 |
| | I5 | #16 |
| Q | J5 | #17 |
| R | K5 | #18 |
| S | I6 | #19 |
| T | J6 | #20 |
| U | K6 | #21 |
| V | I7 | #22 |
| W | J7 | #23 |
| X | K7 | #24 |
| Y | I8 | #25 |
| Z | J8 | #26 |
| | K8 | #27 |
| | I9 | #28 |
| | ≀ | ≀ |
| | K10 | #33 |

Common variables

| |
|---|
| #100 |
| #101 |
| ≀ |
| #149 |

Common variables (Maintenance type)

| |
|---|
| #500 |
| #501 |
| ≀ |
| #999 |

Data input

| |
|---|
| #1000 |
| #1001 |
| ≀ |
| #1015 |
| #1032 |

Data output

| |
|---|
| #1100 |
| #1101 |
| ≀ |
| #1115 |
| #1132 |
| #1133 |

## (2) Series 16-TB/16-TA/18-TB/18-TA/20-TA/21-TB

Tool compensation quantity

| | | |
|---|---|---|
| #2001/2701 ~ #2032/2732 | X. Offset ~ 〃 | 1 ~ 32 |
| #2101/2801 ~ #2132/2832 | Z. Offset ~ 〃 | 1 ~ 32 |
| #2201/2901 ~ #2232/2932 | Tool nose compensation value 〃 ~ 〃 | |
| #2301 ~ #2332 | Hypothetical cutter tip T postion 〃 ~ 〃 | |
| #2401 ~ #2432 | Y. Tool position OFFSET 〃 ~ 〃 | 32 |

* left : Wear compensation
right : Geometry compensation

Work coordinates shift quantity

| | |
|---|---|
| #2501 | X. Work coordinate |
| #2601 | Y. Work coordinate |

G code (G code system A)

| | |
|---|---|
| #4001 | G00–03,32,34,90 |
| #4002 | G96,97 |
| #4003 | * |
| #4004 | G68,69 |
| #4005 | G98,99 |
| #4006 | G20,21 |
| #4007 | G40,41,42 |
| #4008 | G25,26 |
| #4009 | G22,23 |
| #4010 | G80A89 |
| #4011 | * |
| #4012 | G66,67 |
| #4013 | * |
| #4014 | * |
| ~ | ~ |
| #4021 | * |

* : Reserved

Code

| | |
|---|---|
| #4109 | F code |
| #4113 | M code |
| #4114 | Sequence number |
| #4115 | Program number |
| #4119 | S code |
| #4120 | T code |

Position data

| | |
|---|---|
| #5001 | X. Block end position |
| #5002 | Z. 〃 |
| #5003 | 3. 〃 |
| #5004 | 4. 〃 |
| #5021 | X. Machine coordinates |
| #5022 | Z. 〃 |
| #5023 | 3. 〃 |
| #5024 | 4. 〃 |
| #5025 | 5. 〃 |
| #5026 | 6. 〃 |
| #5041 | X. Work coordinates |
| #5042 | Z. 〃 |
| #5043 | 3. 〃 |
| #5044 | 4. 〃 |
| #5045 | 5. 〃 |
| #5046 | 6. 〃 |
| #5061 | X. Skip signal position |
| #5062 | Z. 〃 |
| #5063 | 3. 〃 |
| #5064 | 4. 〃 |
| #5081 | X. Tool compensation value |
| #5082 | Z. 〃 |
| #5101 | X. Servo variation value |
| #5102 | Z. 〃 |
| #5103 | 3. 〃 |
| #5104 | 4. 〃 |

Control variables

| | |
|---|---|
| #3000 | Macro ALARM n<99 |
| #3001 | Clock (msec) |
| #3002 | Clock (hour) |
| #3003 | Single block |
| #3004 | Feed hold |
| #3005 | Setting |
| #3011 | Clock (mm dd yy) |
| #3012 | Clock (hour minute second) |
| #3901 | Number of processed parts |
| #3902 | Number of required parts |

#3003

| n | SBK | FIN |
|---|---|---|
| 0 | ○ | w |
| 1 | × | w |
| 2 | ○ | × |
| 3 | × | × |

○ : No control
× : Control
W : Waiting for help function end signal

#3004

| n | FH | OR | ES |
|---|---|---|---|
| 0 | ○ | ○ | ○ |
| 1 | × | ○ | ○ |
| 2 | ○ | × | ○ |
| 3 | × | × | ○ |
| 4 | ○ | ○ | × |
| 5 | × | ○ | × |
| 6 | ○ | × | × |
| 7 | × | × | × |

FH : Feed hold
OR : Overwrite
ES : Exact stop check

Parameter instructions

| | | |
|---|---|---|
| | | #0 |
| A | A | #1 |
| B | B | #2 |
| C | C | #3 |
| I | I1 | #4 |
| J | J1 | #5 |
| K | K1 | #6 |
| D | I2 | #7 |
| E | J2 | #8 |
| F | K2 | #9 |
| | I3 | #10 |
| H | J3 | #11 |
| | K3 | #12 |
| M | I4 | #13 |
| | J4 | #14 |
| | K4 | #15 |
| | I5 | #16 |
| Q | J5 | #17 |
| R | K5 | #18 |
| S | I6 | #19 |
| T | J6 | #20 |
| U | K6 | #21 |
| V | I7 | #22 |
| W | J7 | #23 |
| X | K7 | #24 |
| Y | I8 | #25 |
| Z | J8 | #26 |
| | K8 | #27 |
| | I9 | #28 |
| | ~ | ~ |
| | K 10 | #33 |

Common variables

| |
|---|
| #100 #101 ~ #149 |

Common variables (Maintenance type)

| |
|---|
| #500 #501 ~ #999 |

Data input

| |
|---|
| #1000 #1001 ~ #1015 |
| #1032 |

Data output

| |
|---|
| #1100 #1101 ~ #1115 |
| #1132 #1133 |

# D MACRO COMPILER/MACRO EXECUTOR WITH CAP I (Series 16/18)

## D.1 OUTLINE

When an option conversational automatic programming function I (CAP I) is selected, the macro program must be loaded in the same ROM module as that in which a custom macro prepared by MTB is loaded.

The display screen and operating processes related to conversational automatic programming cannot be changed by the user's program prepared by the machine tool builder.

In order to store the user's program prepared by the machine tool builder and the CAP I in the same ROM module, the same procedure as making a ROM module shall be performed by compiling the standard format macro-executor program and user's program. However, the dedicated macro executor floppy disk must be used. Specific details are described hereinafter.

**NOTE**

Optional conversational automatic programming function I is not available with Seires 20/21.

## D.2 EQUIPMENT NEEDED FOR COMPILING

In order to produce the ROM for the ROM module in which both the user's program (custom macro) and the CAP I are to be stored, either the P-G Mark II or the P-G Mate are required.

In addition, a dedicated macro compiler system floppy disk is also needed. As the specific system floppy disk to be used depends on the language appearing in the conversational programming displays and corresponding CRT, select the appropriate floppy disk from the table below.

**Table 2 (a) Specifications of FAPT MACRO COMPILER (MACRO EXECUTOR) with CAP I (For Lathe)**

| NC model | P-G model | Name of function | Number of floppy disK File name |
|---|---|---|---|
| FS16-TA | P-G Mark-II P-G Mate | FAPT MACRO COMPILER (MACRO EXECUTOR) 9" High resolution monochrome /color CRT English, Japanese, German, French, Italian, Spanish | A08B-0036-J764#BH01 16-TA, BH01, EXC1/** 16-TA, BH01, EXC2/** 16-TA, CAP, PROG/** |
| | | FAPT MACRO COMPILER (MACRO EXECUTOR) 9" High resolution monochrome /color CRT English, Japanese, Chinese, Korean | A08B-0036-J764#BB02 16-TA, BB02, EXC1/** 16-TA, BB02, EXC2/** 16-TA, CAP, PROG/** |
| FS18-TA | P-G Mark-II P-G Mate | FAPT MACRO COMPILER (MACRO EXECUTOR) 9" High resolution monochrome /color CRT English, Japanese, German, French, Italian, Spanish | A08B-0036-J744#BH51 18-TA, BH51, EXC1/** 18-TA, BH51, EXC2/** 18-TA, CAP, PROG/** |

**NOTE**

The /** at the end of the file name in the table indicates the version number of that system's software.
As differences exist depending on the version number, at the time of actual use, check the file name in the floppy disc and input the appended number.
The 1st file stores files 16-TA.BB**.EXC1 and 16 TA. CAP PROG and 2nd file stores files 16-TA.BB**.EXC2.

**Table 2 (b) Specifications of FAPT MACRO COMPILER (MACRO EXECUTOR) with CAP I (For machining center)**

| NC model | P-G model | Name of function | Number of floppy disK File name |
|---|---|---|---|
| FS16-MA | P-G Mark-II<br>P-G Mate | FAPT MACRO COMPILER (MACRO EXECUTOR)<br>9" High resolution monochrome /color CRT<br>English, Japanese, German, French, Italian, Spanish, Chinese, Korea | A08B-0036-J765#BA21<br>16MA, BA21, EXC1/**<br>16MA, BA21, EXC2/**<br>16MA, CAP, PROG/** |
| FS18-MA | P-G Mark-II<br>P-G Mate | FAPT MACRO COMPILER (MACRO EXECUTOR)<br>9" High resolution monochrome /color CRT<br>English, Japanese, German, French, Italian, Spanish, Chinese, Korea | A08B-0036-J745#BJ51<br>18MA, BA51, EXC1/**<br>18MA, BA51, EXC2/**<br>18MA, CAP, PROG/** |

**NOTE**

The symbol /** appearing at the end of the file name in the table indicates the version number of the system software. As differences exist depending on the version, when putting these to actual use, check the file name in the floppy disk, then input the attached number.

The 1st file stores files 16-MA.BA**.EXC1 and 16MA.CAP PROG and 2nd file stores files 16MA.BA**.EXC2.

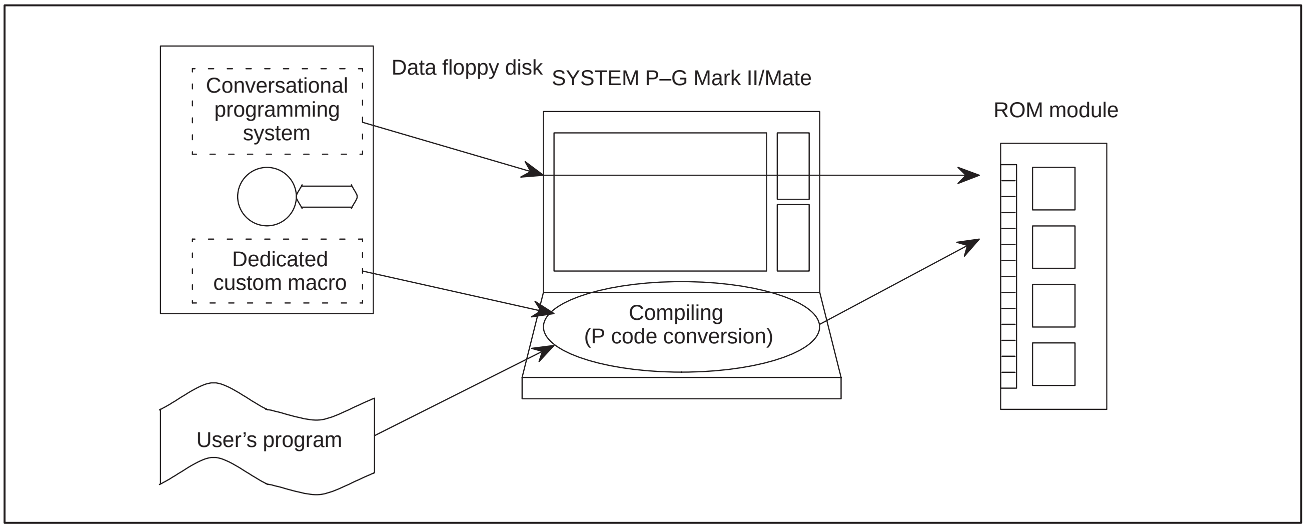
In addition to the above, the following development devices or software are required.

(1) Macro compiler system floppy disk
    P-G Mark-II : A08B-035-J760
    P-G Mate : A08B-036-J760
(2) FA writer : A13B-0157-B001
(3) Adapter : A13B-0157-H010
(4) ROM module (having a memory capacity of 1M byte)
    : A02B-0120-K513

# D.3 COMPILING PROCEDURE USING SYSTEM P-G

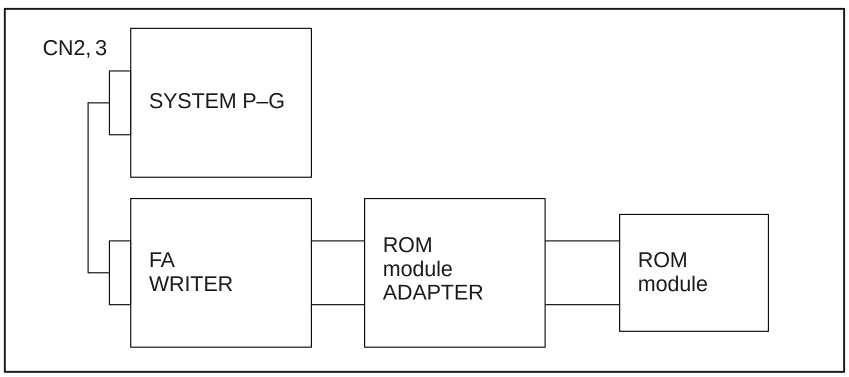
(1) Outline

CAP I consists of a system section and a dedicated custom macro section.The user's program is compiled with this dedicated custom macro section, then stored in the ROM module.




(2) Connection of devices

Connect the FA writer to CN2 or CN3 of P-G. In general, connect CN1 to PPR.




(3) P-G Power Supply ON

Switch the P-G power supply ON.

(4) System loading

1. Insert the FAPT MACRO COMPILER system floppy disk into the disc drive (either one can be used).
2. Depress the [LOAD] button on the upper left of the keyboard for several seconds.
3. Loading is ended when the menu is displayed.

(5) Input of macro executor 1 having CAP

1. Press the R3 key in the initial screen.
2. Key in 'MEXEC IN <NL>' from 'REQUEST='.
3. After the message "Set FD (DATA) and key in 'OK' or 'NO'" is displayed, insert the floppy disc containing macro compiler having CAP into disk drive 0, then key in 'OK @∗∗∗.∗∗∗.EXC1/∗∗<NL>'.

However, as ‘OK @∗∗∗.∗∗∗.EXC1/∗∗<NL>’ is different depending on the macro executor used, refer to the Specifications of FAPT MACRO COMPILER (MACRO EXECU-TER) with CAP in Section 2, then input the file name of the macro executor to be used.

Example)

The file name when using the FS-16MA 9” monochrome high resolution CRT is ”16MA.BA21. EXC1/02”.

Completion of the above-described procedure ends input of macro executor 1 having CAP.

The display returns to the initial screen when <NL> is pressed once.

(6) Input of CAP custom macro program and user’s program.

1. Press the R2 key in the initial screen.
2. With the F2 key in ON status, key in ‘3<NL>’ from ‘NO.=’.
3. After the message ”Set FD (DATA) and key in ‘OK’ or ‘NO’” is displayed, insert the floppy disc containing macro executor having CAP into disk drive 0, then key in ‘OK @∗∗∗∗.CAP.PROG / ∗∗ <NL>’.

   However, as ∗∗∗∗.CAP.PROG / ∗∗ is different depending on the macro executor used, refer to the Table of Specifications of FAPT MACRO COMPILER (MACRO EXECUTOR) with CAP in Section 2, then input the file name of the macro executor to be used.

   Example)

   The file name when using the FS-16MA 9” monochrome high resolution CRT is ”16MA.CAP. PROG/02”.
4. In the case when the program is not completely input to internal memory, external extension is performed. When a message instructing that the floppy disk be set is displayed, insert a blank data floppy disk into disk drive 1, then key in ‘OK<NL>’. This allows input to continue. Do not remove the floppy disk from disk drive 1 during program editing or compiling.
5. When all the dedicated custom macro program input has ended, ‘NO.=’ is displayed. Therefore, if ‘3<NL>’ is keyed in, the same message as in step 3 will be displayed. After inserting the floppy disk containing the user’s program into disk drive 0, key in ‘OK@ file name <NL>’. Specify the file name stored in the user’s program.
6. When all the custom macro program input has ended, ‘NO.=’ is displayed. Pressing <NL> twice will return the display to the intial screen.

(7) Setting compiling parameters

1. Press the R1 key in the initial screen.
2. Key in ‘1<NL>’ from ‘NO.=’.
3. If the <CHG> key is pressed, screen editer will begin, hence the necessary parameters must be set.
4. Setting parameters for FS-16TA.

   8000 00000000

   bit 0 = 1: The program of the block containing M99 ends
   = 0: The program of the block containing M99 does not end (Be sure to set 0)

bit 1 = 1: The program of the block containing M02 ends
= 0: The program of the block containing M99 does not end (Be sure to set 0)

bit 2 = 1: The program of the block containing M30 ends
= 0: The program of the block containing M30 does not end (Be sure to set 0)

bit 7 = 1: The ROM module is used. (For the FS16)
= 0: The ROM cassette is used. (Always specify 1.)

8010 00000001

bit 0 = 1: Corresponds to nacro executor having CAP
= 0: Corresponds to the standard macro executor (Be sure to set 1)

8011 16-TA, ∗∗∗∗. EXC2/∗∗

Specifies the file name of macro executor 2.

However, as ‘16-TA.∗∗∗∗.EXC2/∗∗ <NL>’ is different depending on the macro executor used, refer to the Table of Specifications of FAPT MACRO COMPILER (MACRO EXECUTOR) with CAP in Section 2, then input the file name of the macro executor to be used.

Example)

The file name when using the FS16-TA 9” monochrome high resolution CRT, English, Japanese, German, French, Italian, Spanish version is ”16-TA.BB01.EXC2/07”.

9000 10010000

bit 4 = 1: Allows use of a macro cassette of 1M byte capacity
= 0: Allows use of a macro cassette of other than 1M byte capacity (Be sure to set 1)

bit 7 = 1: Compiles a macro program with custom macro B format
= 0: Compiles a macro program with custom macro A format (Be sure to set 1)

9001 10000001

bit 0 = 1: Outputs sequence number in the P-CODE program at ROM writing
= 0: Does not output sequence number in the P-CODE program at ROM writing (Be sure toset 1)

bit 7 = 1: Can refer to variables at the #10000 level by RAM program
= 0: Cannot refer to variables at the #10000 level by RAM program (Be sure to set 1)

9002 11000000

bit 7 = 1: Macro executor expansion function valid
= 0: Macro executor expansion function not valid (Be sure to set 1)

bit 6 = 1: Automatic work number search valid
= 0: Automatic work number search not valid (Be sure to set 1)

bit 3 = 1: Expanded conversational macro exclusive variables at the #20000 level have non-floating decimal point format

= 0: Expanded conversational macro exclusive variables at the #20000 level have floating decimal point format (Be sure to set 0)

9003 10000000

bit 7 = 1: Special function for CAP valid

= 0: Special function for CAP not valid (Be sure to set 1)

bit 3 = 1: 14” color CRT

= 0: 9” monochrome/color CRT

9007 0100000

bit 6 = 1: FS16-TA CAP I is valid

= 0: FS16-TA CAP I is invalid

9033 97

Dedicated data for CAP

9037 7

Dedicated data for CAP

9044 2044

Dedicated data for CAP Set this value unconditionally for either case of part program memory of 120 m or 320 m.

5. Setting parameters for FS16-MA

8000 10000000

bit 0 = 1: The program of the block containing M99 ends

= 0: The program of the block containing M99 does not end (Be sure to set 0)

bit 1 = 1: The program of the block containing M02 ends

= 0: The program of the block containing M02 does not end (Be sure to set 0)

bit 2 = 1: The program of the block containing M30 ends

= 0: The program of the block containing M30 does not end (Be sure to set 0)

8010 00000001

bit 0 = 1: Corresponds to macro executor having CAP

= 0: Corresponds to the standard macro executor (Be sure to set 1)

8011 16MA, ****, EXC2/**

Specifies the file name of macro executor 2.

However, as ‘16MA.****.EXC2/**’ is different depending on the macro executor used, refer to the Table of Specifications of FAPT MACRO COMPILER (MACRO EXECUTOR) with CAP in Section 2, then input the file name of the macro executor to be used.

Example)

The file name when using the FS16-MA monochrome high resolution CRT is ”16MA.BA21.EXC2/02”.

9000 10010000

bit 4 = 1: Allows use of a macro cassette of 1M Kbyte capacity

= 0: Allows use of a macro cassette of other than 1M Kbyte capacity (Be sure to set 1)

bit 7 = 1: Compiles a macro program with custom macro B format
= 0: Compiles a macro program with custom macro A format (Be sure to set 1)

9001 10000001

bit 0 = 1: Outputs sequence number in the P-CODE program at ROM writing
= 0: Does not output sequence number in the P-CODE program at ROM writing (Be sure to set 1)

bit 7 = 1: Can refer to variables at the #10000 level by RAM program
= 0: Cannot refer to variables at the #10000 level by RAM program (Be sure to set 1)

9002 10000000

bit 7 = 1: Macro executor expansion function valid
= 0: Macro executor expansion function not valid (Be sure to set 1)

bit 6 = 1: Automatic work number search valid
= 0: Automatic work number search not valid

bit 3 = 1: Expanded P-CODE exclusive variables at the #20000 level havenon-floating decimal point format
= 0: Expanded P-CODE exclusive variables at the #20000 level have floating decimal point format

9003 10000000

bit 7 = 1: Special function for CAP valid
= 0: Special function for CAP not valid (Be sure to set 1)

bit 3 = 1: Corresponds to 14" color CRT
= 0: Corresponds to 9" monochrome high resolution CRT

9007 1000000

bit 7 = 1: FS16-MA CAP I is valid
= 0: FS16-MA CAP I is invalid (Always set to 1)

9013 0

9014 0

9015 0

9013 - 9022 0

Be sure to set 0

9033 97

Dedicated data for CAP

9037 10

Dedicated data for CAP

9044 ?

Determine the setting values with reference to the Note) given below.

9045 100

Dedicated data for CAP

9046 900

Dedicated data for CAP

9047 8100

Dedicated data for CAP

**NOTE**

Concerning expanded P-CODE exclusive variables (#2000 -)

If the length of the part program memory is Min. 160m, an arbitrary number of exclusive variables starting from #20000 can be used. Bit 3 of parameter number 9002 allows selection of either floating decimal point format or non-floating decimal point format like regular common variables of these variables.

Parameter No. 9002

bit 3 = 0: Floating decimal point format

= 1: Non-floating decimal point format

The nth value of the number set by parameter number 9044 is the number of expanded P-CODE exclusive variables that can be used.

When the setting value of parameter number 9044 is 0, the expanded P-CODE exclusive variables cannot be used.

When using the floating decimal point format, n=12, and when using the integer format,

n=30.

Example)

Floating decimal point format

#20000 to #20011 can be used when parameter No. 9044=1

#20000 to #20023 can be used when parameter No. 9044=2

Integer format

#20000 to #20029 can be used when parameter No. 9044=1

#20000 to #20059 can be used when parameter No. 9044=2

When using the non-floating decimal point format, values in the range –32768 to 32767 can be set. In the case that assignment is performed at the left side of the assigned statement, settings are made by rounding off to the nearest whole number. In the case that the variable appears in an expression, evaluation is performed after conversion to a floating decimal point format.

Approximately 0.21m of part program memory is used per set of expanded conversational macro exclusive variables (those of parameter no. 9044). Be careful of decrease in the capacity of available tape memory when using a large number of expanded conversational macro exclusive variables. The available part program memory capacity during use in accordance with the number of specified variables is displayed on the program library screen.

The maximum number of variables that can be set to parameter No. 9044 varies, as shown below, in accordance with the part program memory capacity.

Example)

Part program memory 160 m: Parameter No. 9044 = 819

Part program memory 320 m: Parameter No. 9044 = 1638

6. When setting of all necessary parameters is completed, press the R0 key, then press the <NL> key twice to return to the initial screen.

(8) Test Compiling

1. Press the R0 key in the initial screen.
2. To display the source program during compiling, put the F3 key into ON status, and to display only the program number, put the F3 key into OFF status.
3. Key in ‘1<NL>’ from ’NO.=’.
4. As the dedicated custom macro program for CAP and the user’s program are compiled simultaneously, check for the absence of errors.

(9) Setting the FA writer channel

1. Press the R1 key in the initial screen.
2. By keying in ‘2<NL>’ from ’‘NO.=’, the current FA writer channel setting status is displayed.
3. Concerning CN=1

| | |
|---|---|
| ‘ON<NL>’ | :Uses the channel |
| ‘OFF<NL>’ | :Does not use the channel |
| ‘<NL>’ | :Setting is unchanged |

4. The same as above applies to CN=2
5. The same as above applies to CN=3
6. By only pressing ‘<NL>’ in response to ‘NO=’, the display returns to the initial screen.

(10) Preparation of the FA writer

1. Install a ROM module with the previous contents erased in the FA writer
2. Switch ON the power supply of the FA writer
3. Set FA writer to the Remote mode.

(11) ROM writing

1. Press the R0 key in the initial screen.
2. Put the F7 into OFF and the F9 key into ON status. (FA writer selection)
3. When you want to display the source program during compiling, put the F3 key into ON status, and when you want to display only the program number, put the F3 key into OFF status.
4. Key in ‘2<NL>’ from ‘NO=’.
5. If the floppy disk containing macro executor 2 is already inserted in floppy disk drive 0, the data will be read automatically from the second floppy disk and simultaneously written into the ROM module. If the floppy disk has not been inserted in the disk drive,

a message will be displayed. After inserting the macro executor 2 floppy disk into disk drive 0, key ln ‘ OK @ ∗∗∗∗. ∗∗∗∗ EXC2/∗∗ <NL>’.

However, as the above-described ∗∗∗∗. ∗∗∗∗. EXC2/∗∗ is different depending on the macro executor used, refer to the Table of Figure Numbers of FAPT MACRO COMPILER (MACRO EXECUTOR) with CAP in Section 2, then input the file name of the macro executor to be used.

Example)

The file name when using the FS16-TA monochrome high resolution CRT is ”16-TA. BB01. EXC2/07”.

The file name when using the FS16MA 9” monochrome high resolution CRT, English, Japanese, German, French, Italian, Spanish version is ”16-MA.BA01. EXC2/02”.

6. The ROM Module lamp of the FA Writer flicker and writing to the ROM begins.
7. After approximately 1 hour, ROM writing will be completed.

(12) Performance check

After writing is completed, install the macro module into the ”MACRO” socket on the Series 16 main PCB, then check performance.

Custom macro variables #10000 and above and the RAM program are cleared from the setting screen by switching the power supply ON while pressing the [DELETE] button with PWE=1.

## D.4 ROM MODULE

The capacity of the memory area provided for the user's program in combination with the system section of macro executor in the ROM module is 192 Kbyte.

## D.5 ADDITION OF OPTIONS

When the CAP and macro executor function exist in combination as described, procure the options listed below.

(1) FS16-TA CAPI

→ Macro executor provided with CAP I (A02B-0120-J560)

(2) FS16-MA CAP I

→ Macro executor provided with CAP (A02B-0121-J560)

## D.6 LIMITATION ON THE USER'S PROGRAM

Take note that the following limitations exist on the user's program prepared by the machine tool manufacturer (custom macro format).

(1) FS16-TA CAP I

1. Program numbers of 8999 and below cannot be used in the macro program.
2. The only alarm numbers that can be set by the macro program are P/S515 - 519.
3. The only expanded conversational macro variables that can be used are in the range of #10050 - #10399

(2) FS16-MA CAP I

1. Program numbers in the range of 8100 - 8255 as well as 9900 and above cannot be used in the macro program.
2. The only alarm numbers that can be set by the macro program are P/S3000 - 3049.
3. The only expanded conversational macro variables that can be used are #20000 and above.
4. Only G256 to G999 can be used for G-code macro calling, in which a user program calls a P-CODE program. G256 to G999 call programs O8256 to O8999 respectively.
5. When custom macro B is added, G01 to G99 except G65 to G67 (parameters 6050 to 6059) can be used for G-code macro calling.

**NOTE**

1 Conversational programming-related screens cannot be changed by the user's program.

2 Some functions which can be used with the standard macro executor acnnot be used with macro executor provided with CAP.

# E FANUC Super CAP T/CAP II T MACRO COMPILER/EXECUTOR (Series 16/18)

## E.1 GENERAL

FANUC Super Cap T versions 1, 2, and 3 enable machine tool builders to create their own systems. This is done by combining a user program, created by the machine tool builder, and the system program, provided by FANUC, in a single file and storing that file into a custom-built ROM module (versions 1 and 2) or flash ROM (version 3). In other words, some of screens and operations associated with Super CAP T versions 1, 2, and 3 can be created and modified in the user program developed by the machine tool builder.

The user program provided by the machine tool builder and the FANUC system program are both stored into a single ROM module or flash ROM in almost the same way as the standard macro executor system, user program being compiled and stored into a ROM module or flash ROM. Differences between the two methods include the former requiring a special macro executor floppy disk. In comparison with the standard macro executor system, Super CAP T has unique functions and limitations. These are detailed in the subsequent sections.

**NOTE**

1 Whether a function can be used varies from version to version. For details, see the relevant section of this manual.

2 The Series 20/21 does not support a macro compiler/executor with Super CAP T.

## E.2 EQUIPMENT REQUIRED FOR DEVELOPMENT

To develop a user program for use with the macro executor, the equipment listed below needs to be prepared beforehand.

(1) Personal computer (supported by a FAPT macro compiler (for personal computer use))

- Main memory: 640K bytes or more
- OS: MS-DOS (Version 3.1 and later)
- Hard disk with a capacity of about 20M bytes or more (Note 1)
- 5-inch floppy disk drive (Note 2)
- Serial interface
- General-purpose screen editor
- GP-IB interface (Note 3)

**NOTE**

1 The macro compiler utility software and macro executor, when used, need to be stored on hard disk. A macro program to be developed, compile-time list file/object file, link-time ROM-format file, and so forth are output as files onto hard disk.

2 The macro compiler utility software and FANUC Super CAP macro executor system are provided on 5-inch (2HD) floppy disks. A 5-inch floppy disk drive is needed to incorporate the system and executor system.

3 With the FANUC FA-Writer, high-speed ROM write/check operation can be performed via the GP-IB interface.

(2) Peripheral equipment

- FANUC FA Writer (with control software version C or later) (Note 4)
- Memory Card adapter (Note 5)
- Memory Card (greater than 2M bytes) (Note 5)

**NOTE**

1 Required for versions 1 and 2 (FANUC Series 16-T/16-TTA/18-T/18-TTA)

2 Required for version 3 (FANUC Series 16-TB)

(3) FAPT macro compiler (for personal computer use) (A08B-9001-J500#EN03)

**NOTE**

Version 3.1 or later is required for version 3 (FANUC Series 16-TB).

(4) FANUC Super CAP macro executor (Table 1)

**Table 1**

| | NC model | Name of floppy disk, File name |
|---|---|---|
| Ver. 1 | FS16-TA | A08B-9001-J621<br>BH0C_**.EX1, BH0C_**.EX2, BH0C_**.EX3, BH0C_**.SRC |
| | FS16-TTA | A08B-9001-J622<br>BH13_**.EX1, BH13_**.EX2, BH13_**.EX3, BH13_**.SRC<br>BH23_**.EX1, BH23_**.EX2, BH23_**.EX3, BH23_**.SRC |
| | FS18-TA | A08B-9001-J623<br>BH5C_**.EX1, BH5C_**.EX2, BH5C_**.EX3, BH5C_**.SRC |
| Ver. 2 | FS16-TA | A08B-9001-J626<br>BH0D_**.EX1, BH0D_**.EX2, BH0D_**.EX3, BH0D_**.SRC, CAPS_**.SRC |
| | FS16-TTA | A08B-9001-J627<br>BH14_**.EX1, BH14_**.EX2, BH14_**.EX3, BH14_**.SRC<br>BH24_**.EX1, BH24_**.EX2, BH24_**.EX3, BH24_**.SRC, CAPS_**.SRC |
| | FS18-TA | A08B-9001-J628<br>BH5D_**.EX1, BH5D_**.EX2, BH5D_**.EX3, BH5D_**.SRC, CAPS_**.SRC |
| | FS18-TTA | A08B-9001-J629<br>BH64_**.EX1, BH64_**.EX2, BH64_**.EX3, BH64_**.SRC<br>BH74_**.EX1, BH74_**.EX2, BH74_**.EX3, BH74_**.SRC, CAPS_**.SRC |
| Ver. 3 | FS16-TB | A08B-9001-J680<br>BH0F_**.EX1, BH0F_**.EX2, BH0F_**.EX3, BH0F_**.SRC, CAPS_**.SRC |
| | FS16-TB<br>(Two paths) | A08B-9001-J681<br>BH16_**.EX1, BH16_**.EX2, BH16_**.EX3, BH16_**.SRC<br>BH26_**.EX1, BH26_**.EX2, BH26_**.EX3, BH26_**.SRC, CAPS_**.SRC |
| Ver. 4 | 16/18–TC | A08B-9001-J684<br>BH0G_**.EX1, BH0G_**.EX2, BH0G_**.EX3, BH0G_**.SRC |
| | 16/18–TC<br>(Two paths) | A08B-9001-J685<br>BH17_**.EX1, BH17_**.EX2, BH17_**.EX3, BH17_**.SRC,<br>BH27_**.EX1, BH27_**.EX2, BH27_**.EX3, BH27_**.SRC |
| Super<br>CAP II T | 16/18–TC | A08B-9001-J730<br>BH0H_**.EX1, BH0H_**.EX2, BH0H_**.EX3, BH0H_**.SRC |
| | 16/18–TC<br>(Two paths) | A08B-9001-J731<br>BH18_**.EX1, BH18_**.EX2, BH18_**.EX3, BH18_**.SRC,<br>BH28_**.EX1, BH28_**.EX2, BH28_**.EX3, BH28_**.SRC |

**NOTE**

A pair of asterisks (**) appearing at the end of a file name in the table above indicates the version number of that system's software.

(5) Operator's manual of the FAPT macro compiler for personal computer use (B-66102E)

(6) Debug NC unit

## E.3 TYPE OF USER PROGRAMS

With the macro compiler/executor, machine tool builders can generate the types of user programs listed below.

(1) Programs for auxiliary process/transfer process display on the conversational programming menu

For detailed information, see each relevant section that follows.

(2) Programs for auxiliary process/transfer process operation

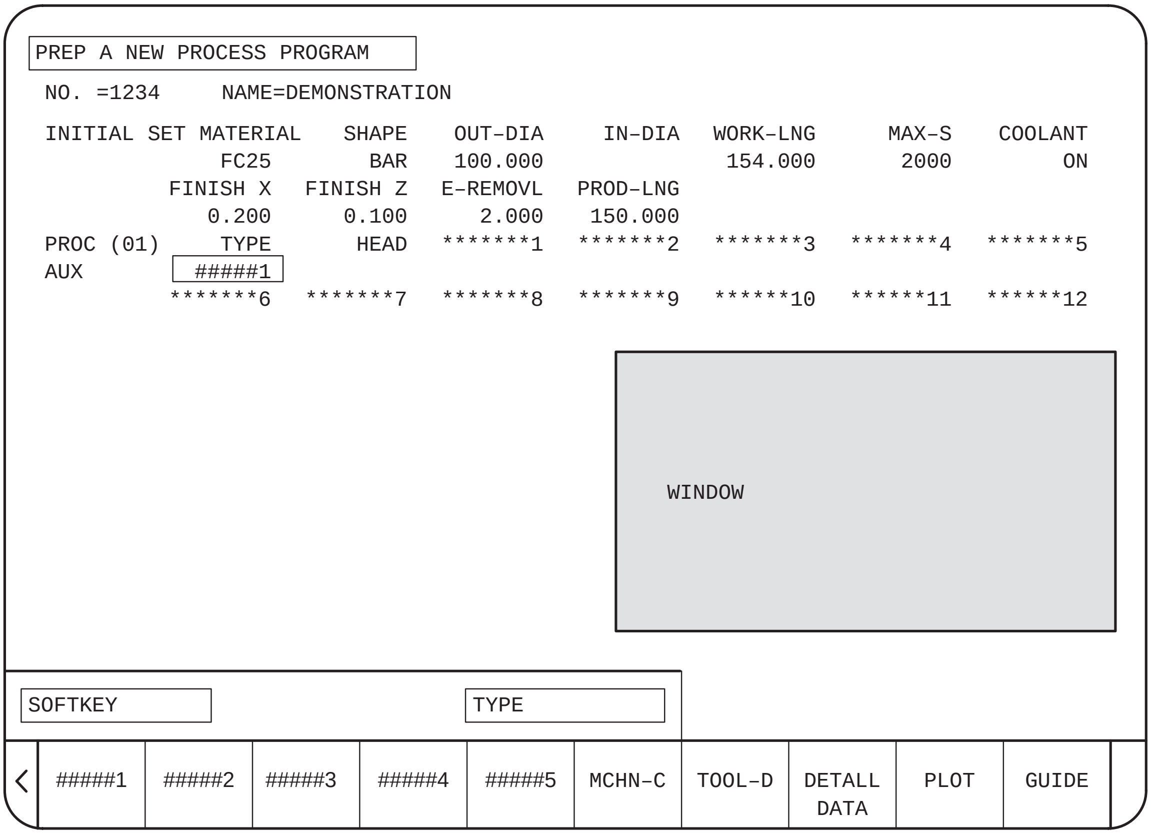
The program format is the same as for an ordinary macro program. For detailed information, see each relevant section that follows.

(3) Programs equivalent to user programs for use with the 0 Series macro compiler/executor

# E.4 DETAILS OF NEW USER PROGRAMS

(1) Programs for auxiliary process/transfer process display on the conversational programming menu

Auxiliary process screen:




For an auxiliary process, programs for directing auxiliary operations such as bar feed, cutting-off, loader operation can be generated in the conversational mode.

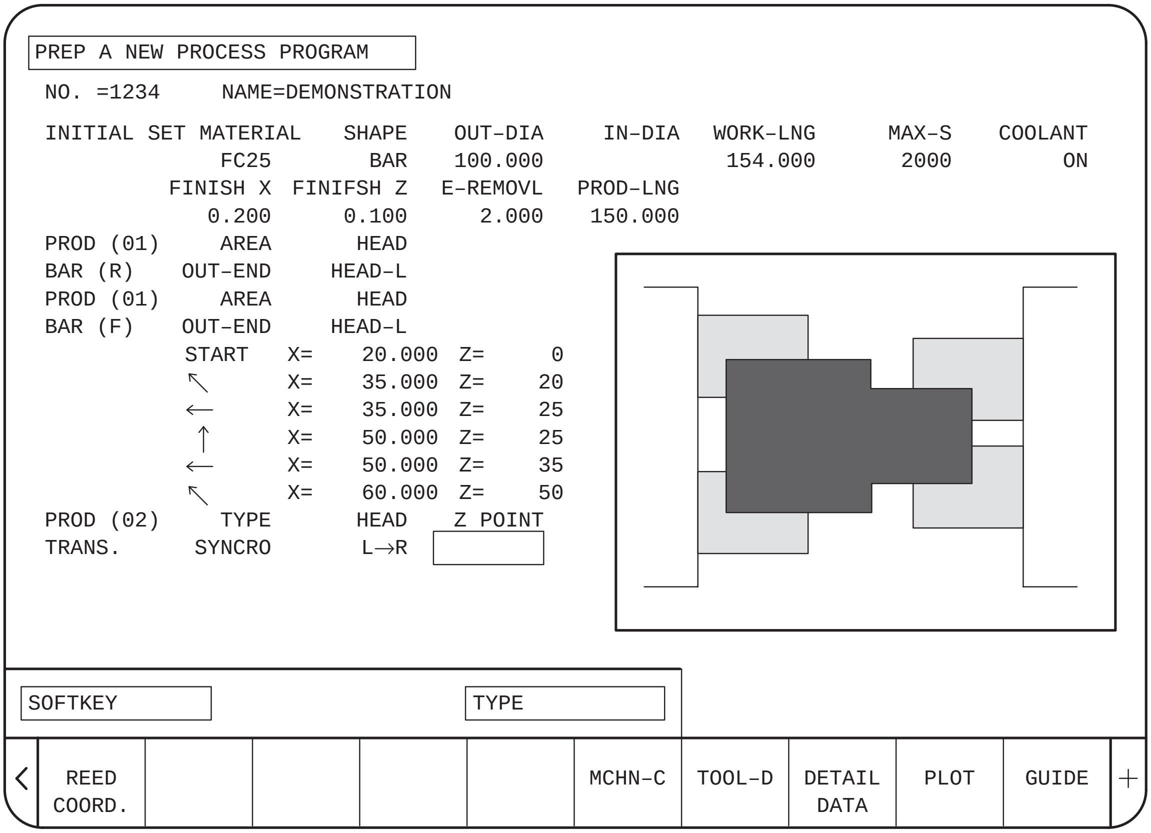
Desired types of auxiliary operation can be selected using the soft keys displayed when the cursor is placed in the data item “TYPE”; up to 20 types of operations can be generated. In this case, no more than five soft keys are displayed at a time. So several groups of soft keys are displayed, group by group, for type selection.

For each type, up to 12 data items (*******1 to ******12 in the figure above) can be displayed. For each data item, five soft keys (#####1 to #####5 in the figure above) can be displayed for operation such as menu selection.

For each data item, detailed menu information including a figure and characters can be displayed in the window (shaded part in the figure above).

The conversational system automatically displays data item names and soft key names if the data item names and soft key names are just stored together with their data identification sequence numbers in a program with a particular program number for the macro executor.

Example of transfer process screen:

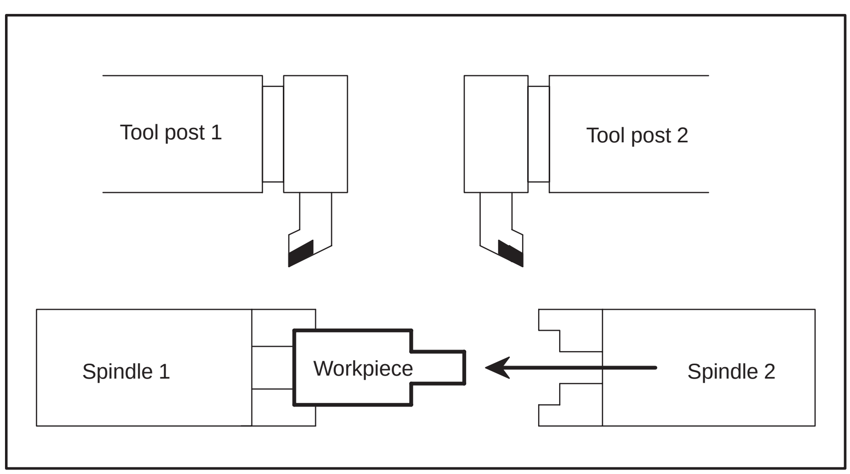



A transfer process is used with a lathe having facing spindles to transfer a workpiece from spindle 1 to spindle 2 or vice versa.

For a transfer process, a user program can display up to 20 types of operations, up to 12 data items for each type, and five soft keys for each data item as in the case of an auxiliary process.

The example above shows the screen for reading, from the machine coordinates, the Z-axis coordinate for transferring a workpiece from spindle 1 to spindle 2.

Example of transferring a workpiece: Workpiece transferred between spindles



The following user programs for auxiliary process/transfer process display are available:

(a) Program for displaying a detailed data screen when the [DETAIL DATA] soft key is pressed

> **NOTE**
> A detailed data screen displays such data that cannot be displayed on a tabular programming screen as shown in the example above.

(b) Program for displaying process data

> **NOTE**
> This program displays a window in multi-window mode, displays soft keys, and controls the cursor.

(c) Program for automatically initializing each data item of a process when the process is to be newly developed

(d) Program executed in moving to another screen

> **NOTE**
> This program is called when the current screen is changed to to another screen such as a current position screen forcibly, for example, by pressing a function key.

## E.5 USER PROGRAM SIZE

An area of 512K bytes is available for the user program used with the macro executor (including the execution macro program of the system).

> **NOTE**
> For the system's execution macro program, 192K bytes are reserved.

# E.6 SOFTWARE PACKAGE FOR USER PROGRAM DEVELOPMENT

## E.6.1 Software Package Associated with Tool Data and Cutting Condition Data

### E.6.1.1 Overview

FANUC Super CAP T versions 1, 2, and 3 can read the tool data, held in a tool data file, into system variables #9900 to #9942. FANUC Super CAP T can also search through the data in the tool data file for a tool ID number.

FANUC Super CAP T versions 2 and 3 can read cutting condition data, held in the cutting condition file, into system variables #9992 o #9996 if ID numbers are already set in system variables #9990 and #9991. (To read the chuck/tailstock data, different variables are used.)

### E.6.1.2 Details of the Software Package Associated with Tool Data

The contents of a tool data file can be determined by setting an ID number, such as the tool ID number of a desired tool, in variable #9999 and reading #9900 to #9942. Tools can also be searched for.

(1) Variables

To macro variables #9900 to 9941, the data (described later) corresponding to each variable is read. The data set in each variable depends on the type of tool. This means that the same variable number may represent different data, depending on the type of tool.

#9999 – An ID number is to be set.

(Tool ID number) :
To read the data of a tool directly by tool ID number, the ID number of the tool is to be set.

(Tool type number) :
To determine the tool ID number of a tool by tool type number, the tool type number of the tool is to be set.
= 100: Outer surface machining
= 150: Inner surface machining
= 200: End facing
= 250: External threading
= 300: Inner threading
= 400: Inner bottom end facing
= 450: Outer surface grooving
= 500: Inner surface grooving
= 550: End face grooving
= 600: Drilling
= 650: Tapping
= 700: Center drilling
= 750: End milling

= 800: Side cutting
= 850: Chamfering
= 900: Reaming
= 950: Boring

(Control symbol number) :

When a tool ID number is determined using tool data, a control symbol number is to be set in order to specify a range of tool data used.
= 11: (Tool file data) < (reference data)
= 12: (Tool file data) ≦ (reference data)
= 13: (Tool file data) = (reference data)
= 14: (Tool file data) ≧ (reference data)
= 15: (Tool file data) > (reference data)

1 : To be set when the data of the tool immediately after the currently selected tool (whose tool ID number is set in #9998) in the tool data list is to be obtained

2 : To be set when the data of the tool immediately before the currently selected tool (whose tool ID number is set in #9998) in the tool data list is to be obtained

3 : To be set when the data of the first tool in the tool data list is to be obtained

–1 : To be set upon completion of reference data input when a tool ID number is to be searched for from tool type information and tool data.

#9998 –

The tool ID number of the currently selected tool (selected using this function immediately before) or the error code ”99” is set. (This variable allows read operation only.)

#9900 – Tool type

= 1 : Outer surface machining
= 2 : Inner surface machining
= 3 : End facing
= 4 : External threading
= 5 : Inner threading

= 7 : Inner bottom end facing
= 8 : Outer surface grooving
= 9 : Inner surface grooving
= 10 : End face grooving
= 11 : Drilling
= 12 : Tapping
= 13 : Center drilling
= 14 : End milling
= 15 : Side cutting
= 16 : Chamfering
= 17 : Reaming
= 18 : Boring

#9901 – Tool direction

= 1 Outer surface or inner surface machining: Right hand
= 2 Outer surface or inner surface machining: Left hand
= 3 End facing: + direction
= 4 End facing: – direction
= 5 Outer or inner surface grooving: Left-reference
= 6 Outer or inner surface grooving: Right-reference
= 7 End face grooving: Down-reference
= 8 End face grooving: Up-reference

= 9 Drilling, tapping, center drilling, end milling, chamfering: End face
= 10 Drilling, tapping, center drilling, end milling, chamfering: Side face
= 11 Outer surface, inner surface, or end face machining: Round nose

#9902 – Rough/finish machining (turning tool)
= 0 : Common = 1 : Rough = 2 : Finish
Rotation/turning (drilling tool)
= 0 : Common = 1 : Rotation = 2 : Turning

#9903 – Specified T code

#9904 – Spindle rotation direction

= 1 : Normal = 2 : Reverse

#9905 – Radius of tool tip, radius of tool

| | General-purpose | Threading | Grooving | Drilling | Tapping | Center drilling |
|---|---|---|---|---|---|---|
| #9906 | Cutting edge angle | _ | Tool length | Tool length | Tool length | Depth of cut |
| #9907 | Tool angle | Tool angle | Cutting edge angle | Point angle | Pitch | Point angle |
| #9908 | Tool width | Tool width | Cutting edge width | Nominal diameter | Nominal diameter | Nominal diameter |

| | End milling | Side cutting | Chamfering | Reaming | Boring |
|---|---|---|---|---|---|
| #9906 | Tool length | Tool length | Edge clearance | Tool length | Tool length |
| #9907 | Number of teeth | Number of teeth | Tool angle | _ | _ |
| #9908 | _ | Tool width | Small diameter | Nominal diameter | Tool width |

#9909 – Virtual tool tip direction

#9910 – Tool material

= 1 : Cemented carbide
= 2 : High-speed tool
= 3 : Special

#9911 – Tool life

#9912 – Spare tool

#9920 - #9927 – Tool tip figure

#9920:X1, #9921:Z1, #9922:X2, #9923:Z2
#9924:X3, #9925:Z3, #9926:X4, #9927:Z4

#9930 - #9941 – Tool shank figure

#9930:X1, #9931:Z1, #9932:X2, #9933:Z2, #9934:X3, #9935:Z3
#9936:X4, #9937:Z4, #9938:X5, #9939:Z5, #9940:X6, #9941:Z6

#9942 – Tool post

= 0 or 1: Tool post 1, = 2: Tool post 2
(Tool post 1 for FANUC Series 16-TA/18-TA)

**NOTE**
System variable #9942 can be used with versions 2 and 3 only.

A user program can use P-CODE variables #10000 to #10699 as desired. With the standard macro compiler/executor (having no conversational functions), the value set in compile parameter 9037 determines the number of P-CODE variables from #10000. In a system that supports the conversational function, however, the parameter is always set to 7.

> **NOTE**
> The P-CODE variables can be extended to #10000 to #13999 by adding the optional 4000 P-CODE variables.

Extended P-CODE variables from #20000 are used by the system. The user program cannot access these variables.

(2) Examples of command format

(a) When the data of a tool is to be referenced based on its tool ID number

```
#9999 = (tool ID number)       ;   – 1.
IF [#9998 EQ 99] GOTO 100      ;   – 2.
#????? = #99??                 ;   – 3.
```

1. Sets the tool ID number of a tool whose data is to be referenced in #9999.
2. Checks #9998 to see if the tool is registered in the tool file.
3. Reads the variables corresponding to data to be referenced if the tool currently selected is registered in the file.

→ If the specified tool ID number cannot be found, 99 is set in #9998.

(b) When the tool ID number and data of the next or previous tool in the tool data list are to be referenced based on the currently selected tool (whose tool ID number is set in #9998)

```
IF [#9998 EQ 99] GOTO 100      ;   – 1.
#????? = #9998                 ;   – 2.
#9999 = 1 or 2                 ;   – 3.
#????? = #99??                 ;   – 4.
```

1. Checks #9998 to see if the currently selected tool is registered.
2. Checks the tool ID number of the currently selected tool.
3. Finds the tool ID number of the next or previous tool by setting 1 or 2 in #9999.
4. Reads the variables corresponding to data to be referenced.

→ If the currently selected tool cannot be found (as in the case of power-up), 99 is set in #9998.

(c) When the tool ID number and data of the first tool in the tool data list are to be referenced

```
#9999 = 3                      ;   – 1.
IF [#9998 EQ 99] GOTO 100      ;   – 2.
#????? = #99??                 ;   – 3.
```

1. Finds the first tool in the tool data list.
2. Checks if the tool is registered.

3. Reads the variables corresponding to data to be referenced if the tool is registered.

→ If the tool cannot be found, 99 is set in #9998.

(d) When a tool ID number is to be searched for using tool type number information and tool data

| | | |
|---|---|---|
| #9999 = (tool type number) | ; | –1. |
| #9999 = (control symbol number) | ; | –2. |
| #99?? = **.* | ; | –3. |
| #9999 = (control symbol number) | ; | –4. |
| #99?? = **.* | ; | –5. |
| #9999 = – 1 | ; | –6. |
| #????? = #9998 | ; | –7. |

1. Sets the tool type number of a tool to be referenced.

2., 4. Writes control symbol numbers specifying an ordinal relationship in #9999.

3., 5. Writes search reference data to the corresponding macro variables.

6. Writes the setting completion code (– 1) to #9999 upon completion of the setting of search reference data.

7. Allows the desired tool ID number to be obtained by reading #9998.

→ If there is no tool that matches specified conditions, 99 is setin #9998.
Control symbol numbers and tool data need to be specified in pairs. Be sure to specify a control symbol number first, then specify tool data. Up to five pairs can be specified. If a control symbol number is missing and the next tool data is read, or tool data is missing and the next control symbol number is read, the error code (99) is set in #9998. When several tools match specified conditions, the tool that has the smallest tool ID number is searched for.

(e) Examples of execution

Assume that the following tools are registered:

| | | | |
|---|---|---|---|
| 101 | Outer surface machining | 451 | Outer surface grooving |
| 102 | Outer surface machining | 452 | Outer surface grooving |
| 103 | Outer surface machining | 453 | Outer surface grooving |
| 104 | Outer surface machining | 601 | Drilling |
| 151 | Inner surface machining | 651 | Tapping |
| 201 | End facing | 701 | Center drilling |
| 251 | External threading | | |

(i) When the data of a certain tool is to be referenced

[The cutting edge angle of the tool ID number 251 (external threading) is read into #10000.]

#9999 = 251 ;

Writes the tool ID number 251 to #9999.

#10000 = #9907 ;

Reads the macro variable, #9907, corresponding to the cutting edge angleof the external threading tool.

(ii) When the data of all tools of a type is to be read

[The tool length data of all outer surface grooving tools is sequentially set in macro variables starting with #10000.]

#9999 = 450 ;

References outer surface grooving tools.

#9999 = – 1 ;

Setting completion code

#10000 = 10000 ;

IF [#9999 EQ 99] GOTO 99 ;

Causes a jump to N99 if there is no outer surface grooving tool.

N10 # [#10000] = #9906 ;

Reads the tool length data of an outer surface grooving tool.

#10000 = #10000+1 ;

#9999 = 1 ;

Searches for the next tool.

IF [#9998 LT 500] GOTO 10 ;

Causes a jump to N10 if there is another outer surface grooving tool.

N99 .....

(iii) When tools satisfying specified conditions are to be searched for

[Outer surface grooving tools that have a tool length of 50mm or more and a tool width of 5 mm or less are searched for.]

| | | |
|---|---|---|
| #9999 = 450 | ; | References outer surface grooving tools. |
| #9999 = 14 | ; | Control symbol number ≧ (50 mm) |
| #9906 = 50000 | ; | Tool length: 50 mm<br>(Least input increment: 0.001 mm) |
| #9999 = 12 | ; | Control symbol number ≦ (5 mm) |
| #9908 = 5000 | ; | Tool width: 5 mm<br>(Least input increment: 0.001 mm) |
| #9999 = – 1 | ; | Setting completion code |

## E.6.1.3 Details of the Software Package Associated with the Cutting Condition Data

If ID numbers are set in system variables #9990 and #9991, the cutting condition data can be read into #9992 to #9996, depending on the values set.

For example, in case (1) (a) shown below, if 18 is set in #9990 and 1 is set in #9991, the feed amount, cutting speed, and cutting depth for roughing material 1 with a general-purpose carbide tool are set in system variables #9992, #9993, and #9994.

> **NOTE**
> 1 Variables #9990 to #9996 are not used to read the chuck/tailstock figure data. (See (10).)
> 2 This function is supported by versions 2 and 3 only.

(1) Cutting condition data screen for general-purpose tools

(a) Carbide tool

```
101  #L   OUTER      T0101  RN   0. 400  AC   95  AN   80  TW   20. 000
102  #R   OUTER      T0202  RN   0. 400  AC   95  AN   55  TW   20. 000
103  #L   OUTER      T0303  RN   0. 400  AC   95  AN   55  TW   20. 000
251  #L   OUTER  TH  T0404  RN   0. 400  AN   60           TW   20. 000
252  #R   OUTER  TH  T0505  RN   0. 400  AN   60           TW   20. 000)

***  MACHINING  CONDITION  (GENERAL)  ***
[#9990=x]                              ROUGHING  | [#9990=y]                  FINISHING
CARBITE                                          | CARBITE
MATERIAL       FEED/REV.  CUT SPD.   CUT DPTH    | MATERIAL       FEED/REV.  CUT SPD.
               (MM/REV.)  (M/MIN.)   (MM)        |                (MM/REV.)  (M/MIN.)
<#9991=1>      #9992      #9993      #9994       | <#9991=1>      #9992      #9993
<#9991=2>      #9992      #9993      #9994       | <#9991=2>      #9992      #9993
<#9991=3>      #9992      #9993      #9994       | <#9991=3>      #9992      #9993
<#9991=4>      #9992      #9993      #9994       | <#9991=4>      #9992      #9993
<#9991=5>      #9992      #9993      #9994       | <#9991=5>      #9992      #9993
<#9991=6>      #9992      #9993      #9994       | <#9991=6>      #9992      #9993
<#9991=7>      #9992      #9993      #9994       | <#9991=7>      #9992      #9993
<#9991=8>      #9992      #9993      #9994       | <#9991=8>      #9992      #9993

                                     SET  FEEDRATE  AMOUNT  BY  MAX.  VALUE.

[ WORK MATERL ] [ ] [ ] [ ] [ ]     [ ] [ ] [ MENU RETRN ] [ ] [ ]
```

- Workpiece material 1 to 8 : x= 18, y= 19
- Workpiece material 9 to 16 : x=155, y=156
- Workpiece material 17 to 24 : x=161, y=162

> **NOTE**
> Workpiece materials 9 to 16 and 17 to 24 are available only when 99 tools can be registered.

(b) High-speed tool

```
 101  #L    OUTER       T0101  RN     0. 400  AC       95  AN       80  TW      20. 000
 102  #R    OUTER       T0202  RN     0. 400  AC       95  AN       55  TW      20. 000
 103  #L    OUTER       T0303  RN     0. 400  AC       95  AN       55  TW      20. 000
 251  #L    OUTER  TH   T0404  RN     0. 400  AN       60                   TW      20. 000
 252  #R    OUTER  TH   T0505  RN     0. 400  AN       60                   TW      20. 000)

 ***  MACHINING  CONDITION  (GENERAL)  ***
 [#9990=x]                           ROUGHING    [#9990=y]                        FINISHING
 HI-SPD                                           HI-SPD
 MATERIAL          FEED/REV. CUT SPD.  CUT DPTH   MATERIAL          FEED/REV. CUT SPD.
                   (MM/REV.) (M/MIN.)  (MM)                         (MM/REV.) (M/MIN.)
 <#9991=1>         #9992     #9993     #9994      <#9991=1>         #9992     #9993
 <#9991=2>         #9992     #9993     #9994      <#9991=2>         #9992     #9993
 <#9991=3>         #9992     #9993     #9994      <#9991=3>         #9992     #9993
 <#9991=4>         #9992     #9993     #9994      <#9991=4>         #9992     #9993
 <#9991=5>         #9992     #9993     #9994      <#9991=5>         #9992     #9993
 <#9991=6>         #9992     #9993     #9994      <#9991=6>         #9992     #9993
 <#9991=7>         #9992     #9993     #9994      <#9991=7>         #9992     #9993
 <#9991=8>         #9992     #9993     #9994      <#9991=8>         #9992     #9993

                                                  SET  FEEDRATE  AMOUNT  BY  MAX.  VALUE.

 |    | WORK  |   |   |   |   |      |   |   | MENU  |   |   |   |
 |    | MATERL|   |   |   |   |      |   |   | RETRN |   |   |   |
```

- Workpiece material 1 to 8 : x= 16, y= 17
- Workpiece material 9 to 16 : x=157, y=158
- Workpiece material 17 to 24 : x=163, y=164

> **NOTE**
> Workpiece materials 9 to 16 and 17 to 24 are available only when 99 tools can be registered.

(c) Special tool

```
 101  #L    OUTER       T0101  RN     0. 400  AC       95  AN       80  TW      20. 000
 102  #R    OUTER       T0202  RN     0. 400  AC       95  AN       55  TW      20. 000
 103  #L    OUTER       T0303  RN     0. 400  AC       95  AN       55  TW      20. 000
 251  #L    OUTER  TH   T0404  RN     0. 400  AN       60                 TW      20. 000
 252  #R    OUTER  TH   T0505  RN     0. 400  AN       60                 TW      20. 000

 ***  MACHINING  CONDITION  (GENERAL)  ***
 [#9990=x]                          ROUGHING      [#9990=y]                       FINISHING
 SPCIAL                                            SPCIAL
 MATERIAL          FEED/REV. CUT SPD.  CUT DPTH    MATERIAL          FEED/REV. CUT SPD.
                   (MM/REV.) (M/MIN.)  (MM)                          (MM/REV.) (M/MIN.)
 <#9991=1>         #9992     #9993     #9994       <#9991=1>         #9992     #9993
 <#9991=2>         #9992     #9993     #9994       <#9991=2>         #9992     #9993
 <#9991=3>         #9992     #9993     #9994       <#9991=3>         #9992     #9993
 <#9991=4>         #9992     #9993     #9994       <#9991=4>         #9992     #9993
 <#9991=5>         #9992     #9993     #9994       <#9991=5>         #9992     #9993
 <#9991=6>         #9992     #9993     #9994       <#9991=6>         #9992     #9993
 <#9991=7>         #9992     #9993     #9994       <#9991=7>         #9992     #9993
 <#9991=8>         #9992     #9993     #9994       <#9991=8>         #9992     #9993

                                                  SET  FEEDRATE  AMOUNT  BY  MAX.  VALUE.

 [WORK MATERL]                                          [MENU RETRN]
```

- Workpiece material 1 to 8 : x= 20, y= 21
- Workpiece material 9 to 16 : x=159, y=160
- Workpiece material 17 to 24 : x=165, y=166

**NOTE**

Workpiece materials 9 to 16 and 17 to 24 are available only when 99 tools can be registered.

(2) Cutting condition data screen for threading tools

(a) Carbide tool

```
101  #L    OUTER       T0101   RN      0. 400  AC       95   AN       80  TW      20. 000
102  #R    OUTER       T0202   RN      0. 400  AC       95   AN       55  TW      20. 000
103  #L    OUTER       T0303   RN      0. 400  AC       95   AN       55  TW      20. 000
251  #L    OUTER  TH   T0404   RN      0. 400  AN       60                TW      20. 000
252  #R    OUTER  TH   T0505   RN      0. 400  AN       60                TW      20. 000

***  MACHINING  CONDITION  (THRZAD)  ***
[#9990=x]
CARBITE
MATERIAL          FEED/REV. CUT SPD.  CUT DPTH
                  (MM/REV.) (M/MIN.)  (MM)
<#9991=1>         #9992     #9993     #9994
<#9991=2>         #9992     #9993     #9994
<#9991=3>         #9992     #9993     #9994
<#9991=4>         #9992     #9993     #9994
<#9991=5>         #9992     #9993     #9994
<#9991=6>         #9992     #9993     #9994
<#9991=7>         #9992     #9993     #9994
<#9991=8>         #9992     #9993     #9994
                                                SET  FEEDRATE  AMOUNT  BY  MAX.  VALUE.

 WORK                                           MENU
 MATERL                                         RETRN
```

- Workpiece material 1 to 8 : x= 97
- Workpiece material 9 to 16 : x=198
- Workpiece material 17 to 24 : x=201

**NOTE**

Workpiece materials 9 to 16 and 17 to 24 are available only when 99 tools can be registered.

(b) High-speed tool

```
101  #L   OUTER     T0101  RN    0. 400  AC    95  AN    80  TW    20. 000
102  #R   OUTER     T0202  RN    0. 400  AC    95  AN    55  TW    20. 000
103  #L   OUTER     T0303  RN    0. 400  AC    95  AN    55  TW    20. 000
251  #L   OUTER  TH T0404  RN    0. 400  AN    60              TW    20. 000
252  #R   OUTER  TH T0505  RN    0. 400  AN    60              TW    20. 000

***  MACHINING  CONDITION  (THRZAD)  ***
[#9990=x]
HI-SPD
MATERIAL          FEED/REV.  CUT SPD.  CUT DPTH
                  (MM/REV.)  (M/MIN.)  (MM)
<#9991=1>         #9992      #9993     #9994
<#9991=2>         #9992      #9993     #9994
<#9991=3>         #9992      #9993     #9994
<#9991=4>         #9992      #9993     #9994
<#9991=5>         #9992      #9993     #9994
<#9991=6>         #9992      #9993     #9994
<#9991=7>         #9992      #9993     #9994
<#9991=8>         #9992      #9993     #9994

                                 SET  FEEDRATE  AMOUNT  BY  MAX.  VALUE.

| WORK   |        |        |        |        |   |        |        | MENU   |        |        |
| MATERL |        |        |        |        |   |        |        | RETRN  |        |        |
```

- Workpiece material 1 to 8 : x= 96
- Workpiece material 9 to 16 : x=197
- Workpiece material 17 to 24 : x=200

**NOTE**

Workpiece materials 9 to 16 and 17 to 24 are available only when 99 tools can be registered.

(c) Special tool

```
101  #L   OUTER      T0101  RN   0. 400  AC   95  AN   80  TW   20. 000
102  #R   OUTER      T0202  RN   0. 400  AC   95  AN   55  TW   20. 000
103  #L   OUTER      T0303  RN   0. 400  AC   95  AN   55  TW   20. 000
251  #L   OUTER  TH  T0404  RN   0. 400  AN   60           TW   20. 000
252  #R   OUTER  TH  T0505  RN   0. 400  AN   60           TW   20. 000

***  MACHINING  CONDITION  (THRZAD)  ***
 [#9990=x]
 SPCIAL
 MATERIAL         FEED/REV. CUT SPD.  CUT DPTH
                  (MM/REV.) (M/MIN.)  (MM)
 <#9991=1>        #9992     #9993     #9994
 <#9991=2>        #9992     #9993     #9994
 <#9991=3>        #9992     #9993     #9994
 <#9991=4>        #9992     #9993     #9994
 <#9991=5>        #9992     #9993     #9994
 <#9991=6>        #9992     #9993     #9994
 <#9991=7>        #9992     #9993     #9994
 <#9991=8>        #9992     #9993     #9994

                                SET  FEEDRATE  AMOUNT  BY  MAX.  VALUE.

     WORK                                  MENU
     MATERL                                RETRN
```

- Workpiece material 1 to 8 : x= 98
- Workpiece material 9 to 16 : x=199
- Workpiece material 17 to 24 : x=202

**NOTE**

Workpiece materials 9 to 16 and 17 to 24 are available only when 99 tools can be registered.

(3) Cutting condition data screen for grooving tools

(a) Carbide tool

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 101 | #L | OUTER | | T0101 | RN | 0. 400 | AC | 95 | AN | 80 | TW | 20. 000 |
| 102 | #R | OUTER | | T0202 | RN | 0. 400 | AC | 95 | AN | 55 | TW | 20. 000 |
| 103 | #L | OUTER | | T0303 | RN | 0. 400 | AC | 95 | AN | 55 | TW | 20. 000 |
| 251 | #L | OUTER | TH | T0404 | RN | 0. 400 | AN | 60 | | | TW | 20. 000 |
| 252 | #R | OUTER | TH | T0505 | RN | 0. 400 | AN | 60 | | | TW | 20. 000 |

*** MACHINING CONDITION (GROOVING) ***

| [#9990=x] | | | ROUGHING |
|---|---|---|---|
| CARBITE | (S-WDTH | MM~ | MM) |
| MATERIAL | FEED/REV. | CUT SPD. | CUT DPTH |
| | (MM/REV.) | (M/MIN.) | (MM) |
| <#9991=1> | #9992 | #9993 | #9994 |
| <#9991=2> | #9992 | #9993 | #9994 |
| <#9991=3> | #9992 | #9993 | #9994 |
| <#9991=4> | #9992 | #9993 | #9994 |
| <#9991=5> | #9992 | #9993 | #9994 |
| <#9991=6> | #9992 | #9993 | #9994 |
| <#9991=7> | #9992 | #9993 | #9994 |
| <#9991=8> | #9992 | #9993 | #9994 |

| [#9990=y] | | FINISHING |
|---|---|---|
| CARBITE | | |
| MATERIAL | FEED/REV. | CUT SPD. |
| | (MM/REV.) | (M/MIN.) |
| <#9991=1> | #9992 | #9993 |
| <#9991=2> | #9992 | #9993 |
| <#9991=3> | #9992 | #9993 |
| <#9991=4> | #9992 | #9993 |
| <#9991=5> | #9992 | #9993 |
| <#9991=6> | #9992 | #9993 |
| <#9991=7> | #9992 | #9993 |
| <#9991=8> | #9992 | #9993 |

SET FEEDRATE AMOUNT BY MAX. VALUE.

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | WORK MATERL | | | | | | | MENU RETRN | | | |

- Workpiece material 1 to 8 : x= 24, y= 25
- Workpiece material 9 to 16 : x=169, y=170
- Workpiece material 17 to 24 : x=175, y=176

**NOTE**

Workpiece materials 9 to 16 and 17 to 24 are available only when 99 tools can be registered.

(b) High-speed tool

```
 101  #L    OUTER      T0101   RN      0. 400  AC        95  AN        80  TW        20. 000
 102  #R    OUTER      T0202   RN      0. 400  AC        95  AN        55  TW        20. 000
 103  #L    OUTER      T0303   RN      0. 400  AC        95  AN        55  TW        20. 000
 251  #L    OUTER  TH  T0404   RN      0. 400  AN        60                TW        20. 000
 252  #R    OUTER  TH  T0505   RN      0. 400  AN        60                TW        20. 000

 ***  MACHINING  CONDITION  (GROOVING)  ***
 [#9990=x]                         ROUGHING     [#9990=y]                      FINISHING
 HI-SPD           (S-WDTH   MM~       MM)       HI-SPD
 MATERIAL         FEED/REV. CUT SPD.  CUT DPTH  MATERIAL          FEED/REV. CUT SPD.
                  (MM/REV.) (M/MIN.)  (MM)                        (MM/REV.) (M/MIN.)
 <#9991=1>        #9992     #9993     #9994     <#9991=1>         #9992     #9993
 <#9991=2>        #9992     #9993     #9994     <#9991=2>         #9992     #9993
 <#9991=3>        #9992     #9993     #9994     <#9991=3>         #9992     #9993
 <#9991=4>        #9992     #9993     #9994     <#9991=4>         #9992     #9993
 <#9991=5>        #9992     #9993     #9994     <#9991=5>         #9992     #9993
 <#9991=6>        #9992     #9993     #9994     <#9991=6>         #9992     #9993
 <#9991=7>        #9992     #9993     #9994     <#9991=7>         #9992     #9993
 <#9991=8>        #9992     #9993     #9994     <#9991=8>         #9992     #9993

                                                 SET  FEEDRATE  AMOUNT  BY  MAX.  VALUE.

 |    | WORK  |     |     |     |     |   |     |     | MENU  |     |     |    |
 |    | MATERL|     |     |     |     |   |     |     | RETRN |     |     |    |
```

- Workpiece material 1 to 8 : x= 22, y= 23
- Workpiece material 9 to 16 : x=167, y=168
- Workpiece material 17 to 24 : x=173, y=174

**NOTE**

Workpiece materials 9 to 16 and 17 to 24 are available only when 99 tools can be registered.

(c) Special tool

```
101  #L   OUTER       T0101  RN    0. 400  AC     95  AN     80  TW    20. 000
102  #R   OUTER       T0202  RN    0. 400  AC     95  AN     55  TW    20. 000
103  #L   OUTER       T0303  RN    0. 400  AC     95  AN     55  TW    20. 000
251  #L   OUTER  TH   T0404  RN    0. 400  AN     60                TW    20. 000
252  #R   OUTER  TH   T0505  RN    0. 400  AN     60                TW    20. 000

***  MACHINING  CONDITION  (GROOVING)  ***
[#9990=x]                       ROUGHING    [#9990=y]                 FINISHING
SPCIAL          (S-WDTH   MM~      MM)       SPCIAL
MATERIAL        FEED/REV. CUT SPD. CUT DPTH  MATERIAL        FEED/REV. CUT SPD.
                (MM/REV.) (M/MIN.) (MM)                      (MM/REV.) (M/MIN.)
<#9991=1>       #9992     #9993    #9994     <#9991=1>       #9992     #9993
<#9991=2>       #9992     #9993    #9994     <#9991=2>       #9992     #9993
<#9991=3>       #9992     #9993    #9994     <#9991=3>       #9992     #9993
<#9991=4>       #9992     #9993    #9994     <#9991=4>       #9992     #9993
<#9991=5>       #9992     #9993    #9994     <#9991=5>       #9992     #9993
<#9991=6>       #9992     #9993    #9994     <#9991=6>       #9992     #9993
<#9991=7>       #9992     #9993    #9994     <#9991=7>       #9992     #9993
<#9991=8>       #9992     #9993    #9994     <#9991=8>       #9992     #9993
                                     SET  FEEDRATE  AMOUNT  BY  MAX.  VALUE.

| WORK   |  |  |  |  |      |  |  | MENU  |  |  |
| MATERL |  |  |  |  |      |  |  | RETRN |  |  |
```

- Workpiece material 1 to 8 : x= 26, y= 27
- Workpiece material 9 to 16 : x=171, y=172
- Workpiece material 17 to 24 : x=177, y=178

**NOTE**
Workpiece materials 9 to 16 and 17 to 24 are available only when 99 tools can be registered.

(4) Cutting condition data screen for drilling tools

(a) High-speed tool

```
101  #L   OUTER      T0101  RN    0. 400  AC     95  AN     80  TW     20. 000
102  #R   OUTER      T0202  RN    0. 400  AC     95  AN     55  TW     20. 000
103  #L   OUTER      T0303  RN    0. 400  AC     95  AN     55  TW     20. 000
251  #L   OUTER  TH  T0404  RN    0. 400  AN     60                TW     20. 000
252  #R   OUTER  TH  T0505  RN    0. 400  AN     60                TW     20. 000

***  MACHINING  CONDITION  (DRILLING)  ***
[#9990=x]
HI-SPD  DRIL       (S-DIA.   MM~       MM)
MATERIAL           FEED/REV. CUT SPD.
                   (MM/REV.) (M/MIN.)
<#9991=1>          #9992     #9993
<#9991=2>          #9992     #9993
<#9991=3>          #9992     #9993
<#9991=4>          #9992     #9993
<#9991=5>          #9992     #9993
<#9991=6>          #9992     #9993
<#9991=7>          #9992     #9993
<#9991=8>          #9992     #9993

                         SET  FEEDRATE  AMOUNT  BY  MAX.  VALUE.

| WORK  | DRILL | CENTER | REAMER | BORE |   |   | MENU  |   |   |
| MATERL|       |        |        |      |   |   | RETRN |   |   |
```

<DRILL>

- Workpiece material 1 to 8 : x=28
- Workpiece material 9 to 16 : x=179
- Workpiece material 17 to 24 : x=182

<CENTER>

- Workpiece material 1 to 8 : x=92
- Workpiece material 9 to 16 : x=185
- Workpiece material 17 to 24 : x=188

<REAMER>

- Workpiece material 1 to 8 : x=137
- Workpiece material 9 to 16 : x=140
- Workpiece material 17 to 24 : x=143

<BORE>

- Workpiece material 1 to 8 : x=146
- Workpiece material 9 to 16 : x=149
- Workpiece material 17 to 24 : x=152

**NOTE**

Workpiece materials 9 to 16 and 17 to 24 are available only when 99 tools can be registered.

(b) Carbide tool

```
101  #L   OUTER      T0101  RN    0. 400  AC     95  AN     80  TW     20. 000
102  #R   OUTER      T0202  RN    0. 400  AC     95  AN     55  TW     20. 000
103  #L   OUTER      T0303  RN    0. 400  AC     95  AN     55  TW     20. 000
251  #L   OUTER  TH  T0404  RN    0. 400  AN     60                TW     20. 000
252  #R   OUTER  TH  T0505  RN    0. 400  AN     60                TW     20. 000

 ***  MACHINING  CONDITION  (DRILLING)  ***
[#9990=x]
CARBID  DRIL     (S-DIA.   MM~       MM)
MATERIAL         FEED/REV. CUT SPD.
                 (MM/REV.) (M/MIN.)
<#9991=1>        #9992     #9993
<#9991=2>        #9992     #9993
<#9991=3>        #9992     #9993
<#9991=4>        #9992     #9993
<#9991=5>        #9992     #9993
<#9991=6>        #9992     #9993
<#9991=7>        #9992     #9993
<#9991=8>        #9992     #9993

                                  SET  FEEDRATE  AMOUNT  BY  MAX.  VALUE.

| WORK  | DRILL | CENTER | REAMER | BORE |  |  | MENU  |  |  |
| MATERL|       |        |        |      |  |  | RETRN |  |  |
```

<DRILL>

- Workpiece material 1 to 8 : x= 29
- Workpiece material 9 to 16 : x=180
- Workpiece material 17 to 24 : x=183

<CENTER>

- Workpiece material 1 to 8 : x= 93
- Workpiece material 9 to 16 : x=186
- Workpiece material 17 to 24 : x=189

<REAMER>

- Workpiece material 1 to 8 : x=138
- Workpiece material 9 to 16 : x=141
- Workpiece material 17 to 24 : x=144

<BORE>

- Workpiece material 1 to 8 : x=147
- Workpiece material 9 to 16 : x=150
- Workpiece material 17 to 24 : x=153

**NOTE**

Workpiece materials 9 to 16 and 17 to 24 are available only when 99 tools can be registered.

(c) Special tool

```
101  #L   OUTER       T0101  RN    0. 400  AC    95  AN    80  TW    20. 000
102  #R   OUTER       T0202  RN    0. 400  AC    95  AN    55  TW    20. 000
103  #L   OUTER       T0303  RN    0. 400  AC    95  AN    55  TW    20. 000
251  #L   OUTER  TH   T0404  RN    0. 400  AN    60              TW    20. 000
252  #R   OUTER  TH   T0505  RN    0. 400  AN    60              TW    20. 000

 ***  MACHINING  CONDITION  (DRILLING)  ***
 [#9990=x]
 SPCIAL  DRIL       (S-DIA.   MM~       MM)
 MATERIAL           FEED/REV. CUT SPD.
                    (MM/REV.) (M/MIN.)
 <#9991=1>          #9992     #9993
 <#9991=2>          #9992     #9993
 <#9991=3>          #9992     #9993
 <#9991=4>          #9992     #9993
 <#9991=5>          #9992     #9993
 <#9991=6>          #9992     #9993
 <#9991=7>          #9992     #9993
 <#9991=8>          #9992     #9993
                                     SET  FEEDRATE  AMOUNT  BY  MAX.  VALUE.

| WORK  | DRILL | CENTER | REAMER | BORE |  |  |  | MENU  |  |  |
| MATERL|       |        |        |      |  |  |  | RETRN |  |  |
```

<DRILL>

- Workpiece material 1 to 8 : x= 30
- Workpiece material 9 to 16 : x=181
- Workpiece material 17 to 24 : x=184

<CENTER>

- Workpiece material 1 to 8 : x= 94
- Workpiece material 9 to 16 : x=187
- Workpiece material 17 to 24 : x=190

<REAMER>

- Workpiece material 1 to 8 : x=139
- Workpiece material 9 to 16 : x=142
- Workpiece material 17 to 24 : x=145

<BORE>

- Workpiece material 1 to 8 : x=148
- Workpiece material 9 to 16 : x=151
- Workpiece material 17 to 24 : x=154

**NOTE**

Workpiece materials 9 to 16 and 17 to 24 are available only when 99 tools can be registered.

(5) Cutting condition data screen for taps

(a) High-speed tool

```
101  #L   OUTER       T0101  RN   0. 400  AC   95  AN   80  TW   20. 000
102  #R   OUTER       T0202  RN   0. 400  AC   95  AN   55  TW   20. 000
103  #L   OUTER       T0303  RN   0. 400  AC   95  AN   55  TW   20. 000
251  #L   OUTER  TH   T0404  RN   0. 400  AN   60               TW   20. 000
252  #R   OUTER  TH   T0505  RN   0. 400  AN   60               TW   20. 000

 ***  MACHINING  CONDITION  (TAPPING)  ***
 [#9990=x]
 HI-SPD  TAP       (S-DIA.   MM~       MM)
 MATERIAL          CUT  SPD.
                   (M/MIN.)
 <#9991=1>         #9992
 <#9991=2>         #9992
 <#9991=3>         #9992
 <#9991=4>         #9992
 <#9991=5>         #9992
 <#9991=6>         #9992
 <#9991=7>         #9992
 <#9991=8>         #9992
                                   SET  FEEDRATE  AMOUNT  BY  MAX.  VALUE.

 |  | WORK   |  |  |  |  |    |  |  | MENU  |  |  |  |
 |  | MATERL |  |  |  |  |    |  |  | RETRN |  |  |  |
```

<TAP>

- Workpiece material 1 to 8 : x= 60
- Workpiece material 9 to 16 : x=191
- Workpiece material 17 to 24 : x=194

> **NOTE**
>
> Workpiece materials 9 to 16 and 17 to 24 are available only when 99 tools can be registered.

(b) Carbide tool

```
101  #L   OUTER      T0101  RN    0. 400  AC    95  AN    80  TW    20. 000
102  #R   OUTER      T0202  RN    0. 400  AC    95  AN    55  TW    20. 000
103  #L   OUTER      T0303  RN    0. 400  AC    95  AN    55  TW    20. 000
251  #L   OUTER  TH  T0404  RN    0. 400  AN    60              TW    20. 000
252  #R   OUTER  TH  T0505  RN    0. 400  AN    60              TW    20. 000

***  MACHINING  CONDITION  (TAPPING)  ***
[#9990=x]
CARBID  TAP       (S-DIA.   MM~       MM)
MATERIAL          CUT  SPD.
                  (M/MIN.)
<#9991=1>         #9992
<#9991=2>         #9992
<#9991=3>         #9992
<#9991=4>         #9992
<#9991=5>         #9992
<#9991=6>         #9992
<#9991=7>         #9992
<#9991=8>         #9992
                                          SET  FEEDRATE  AMOUNT  BY  MAX.  VALUE.

|     | WORK   |     |     |     |     |  |     |     | MENU  |     |     |     |
|     | MATERL |     |     |     |     |  |     |     | RETRN |     |     |     |
```

<TAP>

- Workpiece material 1 to 8 : x= 61
- Workpiece material 9 to 16 : x=192
- Workpiece material 17 to 24 : x=195

**NOTE**

Workpiece materials 9 to 16 and 17 to 24 are available only when 99 tools can be registered.

(c) Special tool

```
101  #L   OUTER        T0101  RN    0. 400  AC      95  AN      80  TW     20. 000
102  #R   OUTER        T0202  RN    0. 400  AC      95  AN      55  TW     20. 000
103  #L   OUTER        T0303  RN    0. 400  AC      95  AN      55  TW     20. 000
251  #L   OUTER  TH    T0404  RN    0. 400  AN      60                 TW     20. 000
252  #R   OUTER  TH    T0505  RN    0. 400  AN      60                 TW     20. 000

*** MACHINING CONDITION (TAPPING) ***
[#9990=x]
SPECIAL TAP       (S-DIA.   MM~      MM)
MATERIAL          CUT SPD.
                  (M/MIN.)
<#9991=1>         #9992
<#9991=2>         #9992
<#9991=3>         #9992
<#9991=4>         #9992
<#9991=5>         #9992
<#9991=6>         #9992
<#9991=7>         #9992
<#9991=8>         #9992
                                   SET FEEDRATE AMOUNT BY MAX. VALUE.

| WORK   |                          | MENU  |
| MATERL |                          | RETRN |
```

<TAP>

- Workpiece material 1 to 8 : x= 62
- Workpiece material 9 to 16 : x=193
- Workpiece material 17 to 24 : x=196

**NOTE**

Workpiece materials 9 to 16 and 17 to 24 are available only when 99 tools can be registered.

(6) Cutting condition data screen for rotary tools

(a) End mill

(i) High-speed tool

```
 101  #L    OUTER      T0101  RN     0. 400  AC      95  AN      80  TW      20. 000
 102  #R    OUTER      T0202  RN     0. 400  AC      95  AN      55  TW      20. 000
 103  #L    OUTER      T0303  RN     0. 400  AC      95  AN      55  TW      20. 000
 251  #L    OUTER  TH  T0404  RN     0. 400  AN      60              TW      20. 000
 252  #R    OUTER  TH  T0505  RN     0. 400  AN      60              TW      20. 000

*** MACHINING CONDITION (ENDMILL) ***
[#9990=x]                             ROUGHING   [#9990=y]                     FINISHING
HI-SPDENDMIL  (S-WDTH   MM~       MM)            HI-SPDENDMIL
MATERIAL      FEED QUT-H FEED QNT-A CUT SPD.     MATERIAL     FEED QUT-H FEED QNT-A CUT SPD.
              (MM/EDG.) (M/EDG.)  (M/MIN.)                    (MM/EDG.)  (M/EDG.)  (M/MIN.)
<#9991=1>     #9992     #9993     #9994          <#9991=1>    #9992      #9993     #9994
<#9991=2>     #9992     #9993     #9994          <#9991=2>    #9992      #9993     #9994
<#9991=3>     #9992     #9993     #9994          <#9991=3>    #9992      #9993     #9994
<#9991=4>     #9992     #9993     #9994          <#9991=4>    #9992      #9993     #9994
<#9991=5>     #9992     #9993     #9994          <#9991=5>    #9992      #9993     #9994
<#9991=6>     #9992     #9993     #9994          <#9991=6>    #9992      #9993     #9994
<#9991=7>     #9992     #9993     #9994          <#9991=7>    #9992      #9993     #9994
<#9991=8>     #9992     #9993     #9994          <#9991=8>    #9992      #9993     #9994
                                                 SET  FEEDRATE  AMOUNT  BY  MAX.  VALUE.

|   | WORK   | ENDMIL | SIDCUT | CHAMFR |   |  |   |   | MENU  |   |   |   |
|   | MATERL |        |        |        |   |  |   |   | RETRN |   |   |   |
```

- Workpiece material 1 to 8 : x=9003, y=9004
- Workpiece material 9 to 16 : x=9057, y=9058
- Workpiece material 17 to 24 : x=9063, y=9064

**NOTE**

Workpiece materials 9 to 16 and 17 to 24 are available only when 99 tools can be registered.

(ii) Carbide tool

```
 101  #L   OUTER       T0101   RN     0. 400  AC      95  AN      80  TW     20. 000
 102  #R   OUTER       T0202   RN     0. 400  AC      95  AN      55  TW     20. 000
 103  #L   OUTER       T0303   RN     0. 400  AC      95  AN      55  TW     20. 000
 251  #L   OUTER  TH   T0404   RN     0. 400  AN      60              TW     20. 000
 252  #R   OUTER  TH   T0505   RN     0. 400  AN      60              TW     20. 000

***  MACHINING  CONDITION  (ENDMILL)  ***
[#9990=x]                                ROUGHING    [#9990=y]                        FINISHING
CARBID ENDMIL (S-WDTH   MM~       MM)                CARBID ENDMIL
MATERIAL      FEED  QUT-H FEED QNT-A CUT SPD.        MATERIAL     FEED  QUT-H FEED  QNT-A CUT SPD.
              (MM/EDG.) (M/EDG.)  (M/MIN.)                        (MM/EDG.)  (M/EDG.)  (M/MIN.)
<#9991=1>     #9992     #9993     #9994              <#9991=1>    #9992      #9993     #9994
<#9991=2>     #9992     #9993     #9994              <#9991=2>    #9992      #9993     #9994
<#9991=3>     #9992     #9993     #9994              <#9991=3>    #9992      #9993     #9994
<#9991=4>     #9992     #9993     #9994              <#9991=4>    #9992      #9993     #9994
<#9991=5>     #9992     #9993     #9994              <#9991=5>    #9992      #9993     #9994
<#9991=6>     #9992     #9993     #9994              <#9991=6>    #9992      #9993     #9994
<#9991=7>     #9992     #9993     #9994              <#9991=7>    #9992      #9993     #9994
<#9991=8>     #9992     #9993     #9994              <#9991=8>    #9992      #9993     #9994
                                                     SET  FEEDRATE  AMOUNT  BY  MAX.  VALUE.

|  | WORK   | ENDMIL | SIDCUT | CHAMFR |  |  |  | MENU  |  |  |  |
|  | MATERL |        |        |        |  |  |  | RETRN |  |  |  |
```

- Workpiece material 1 to 8 : x=9001, y=9002
- Workpiece material 9 to 16 : x=9055, y=9056
- Workpiece material 17 to 24 : x=9061, y=9062

**NOTE**

Workpiece materials 9 to 16 and 17 to 24 are available only when 99 tools can be registered.

(iii) Special tool

```
 101  #L   OUTER      T0101  RN    0. 400  AC     95  AN     80  TW     20. 000
 102  #R   OUTER      T0202  RN    0. 400  AC     95  AN     55  TW     20. 000
 103  #L   OUTER      T0303  RN    0. 400  AC     95  AN     55  TW     20. 000
 251  #L   OUTER  TH  T0404  RN    0. 400  AN     60                TW     20. 000
 252  #R   OUTER  TH  T0505  RN    0. 400  AN     60                TW     20. 000

***  MACHINING  CONDITION  (ENDMILL)  ***
[#9990=x]                           ROUGHING    [#9990=y]                   FINISHING
SPCIAL ENDMIL (S-WDTH   MM~         MM)         SPCIAL ENDMIL
MATERIAL      FEED QUT-H FEED QNT-A CUT SPD.    MATERIAL     FEED QUT-H FEED QNT-A CUT SPD.
              (MM/EDG.)  (M/EDG.)   (M/MIN.)                 (MM/EDG.)  (M/EDG.)   (M/MIN.)
<#9991=1>     #9992      #9993      #9994       <#9991=1>    #9992      #9993      #9994
<#9991=2>     #9992      #9993      #9994       <#9991=2>    #9992      #9993      #9994
<#9991=3>     #9992      #9993      #9994       <#9991=3>    #9992      #9993      #9994
<#9991=4>     #9992      #9993      #9994       <#9991=4>    #9992      #9993      #9994
<#9991=5>     #9992      #9993      #9994       <#9991=5>    #9992      #9993      #9994
<#9991=6>     #9992      #9993      #9994       <#9991=6>    #9992      #9993      #9994
<#9991=7>     #9992      #9993      #9994       <#9991=7>    #9992      #9993      #9994
<#9991=8>     #9992      #9993      #9994       <#9991=8>    #9992      #9993      #9994
                                                SET  FEEDRATE  AMOUNT  BY  MAX.  VALUE.

| WORK   | ENDMIL | SIDCUT | CHAMFR |  |  |  | MENU  |  |  |
| MATERL |        |        |        |  |  |  | RETRN |  |  |
```

- Workpiece material 1 to 8 : x=9005, y=9006
- Workpiece material 9 to 16 : x=9059, y=9060
- Workpiece material 17 to 24 : x=9065, y=9066

**NOTE**

Workpiece materials 9 to 16 and 17 to 24 are available only when 99 tools can be registered.

(b) Side cutter

(i) High-speed tool

```
 101  #L    OUTER       T0101   RN      0. 400  AC       95  AN       80  TW       20. 000
 102  #R    OUTER       T0202   RN      0. 400  AC       95  AN       55  TW       20. 000
 103  #L    OUTER       T0303   RN      0. 400  AC       95  AN       55  TW       20. 000
 251  #L    OUTER  TH   T0404   RN      0. 400  AN       60                TW       20. 000
 252  #R    OUTER  TH   T0505   RN      0. 400  AN       60                TW       20. 000

 ***  MACHINING  CONDITION  (SIDECUT)  ***
 [#9990=x]                            ROUGHING      [#9990=y]                 FINISHING
 HI-SPD SIDECUT(S-WDTH   MM~       MM)            HI-SPD SIDECUT
 MATERIAL      FEED/REV. CUT SPD.                 MATERIAL     FEED/REV.  CUT SPD.
               (MM/EDG.) (M/MIN.)                              (MM/EDG.)  (M/MIN.)
 <#9991=1>     #9992     #9993                    <#9991=1>    #9992      #9993
 <#9991=2>     #9992     #9993                    <#9991=2>    #9992      #9993
 <#9991=3>     #9992     #9993                    <#9991=3>    #9992      #9993
 <#9991=4>     #9992     #9993                    <#9991=4>    #9992      #9993
 <#9991=5>     #9992     #9993                    <#9991=5>    #9992      #9993
 <#9991=6>     #9992     #9993                    <#9991=6>    #9992      #9993
 <#9991=7>     #9992     #9993                    <#9991=7>    #9992      #9993
 <#9991=8>     #9992     #9993                    <#9991=8>    #9992      #9993

                                                  SET  FEEDRATE  AMOUNT  BY  MAX.  VALUE.

 |    | WORK   | ENDMIL | SIDCUT | CHAMFR |        |  |        |        | MENU   |        |        |    |
 |    | MATERL |        |        |        |        |  |        |        | RETRN  |        |        |    |
```

- Workpiece material 1 to 8 : x=9009, y=9010
- Workpiece material 9 to 16 : x=9069, y=9070
- Workpiece material 17 to 24 : x=9075, y=9076

**NOTE**

Workpiece materials 9 to 16 and 17 to 24 are available only when 99 tools can be registered.

(ii) Carbide tool

```
101  #L   OUTER      T0101  RN   0. 400  AC   95  AN   80  TW   20. 000
102  #R   OUTER      T0202  RN   0. 400  AC   95  AN   55  TW   20. 000
103  #L   OUTER      T0303  RN   0. 400  AC   95  AN   55  TW   20. 000
251  #L   OUTER  TH  T0404  RN   0. 400  AN   60               TW   20. 000
252  #R   OUTER  TH  T0505  RN   0. 400  AN   60               TW   20. 000

***  MACHINING  CONDITION  (SIDECUT)  ***
[#9990=x]                         ROUGHING     [#9990=y]                 FINISHING
CARBID SIDECUT(S-WDTH   MM~       MM)          CARBID SIDECUT
MATERIAL       FEED/REV.  CUT SPD.             MATERIAL     FEED/REV.  CUT SPD.
               (MM/EDG.)  (M/MIN.)                          (MM/EDG.)  (M/MIN.)
<#9991=1>      #9992      #9993                <#9991=1>    #9992      #9993
<#9991=2>      #9992      #9993                <#9991=2>    #9992      #9993
<#9991=3>      #9992      #9993                <#9991=3>    #9992      #9993
<#9991=4>      #9992      #9993                <#9991=4>    #9992      #9993
<#9991=5>      #9992      #9993                <#9991=5>    #9992      #9993
<#9991=6>      #9992      #9993                <#9991=6>    #9992      #9993
<#9991=7>      #9992      #9993                <#9991=7>    #9992      #9993
<#9991=8>      #9992      #9993                <#9991=8>    #9992      #9993
                                               SET  FEEDRATE  AMOUNT  BY  MAX.  VALUE.

|      | WORK   | ENDMIL | SIDCUT | CHAMFR |      |      |      | MENU  |      |      |      |
|      | MATERL |        |        |        |      |      |      | RETRN |      |      |      |
```

- Workpiece material 1 to 8 : x=9007, y=9008
- Workpiece material 9 to 16 : x=9067, y=9068
- Workpiece material 17 to 24 : x=9073, y=9074

**NOTE**

Workpiece materials 9 to 16 and 17 to 24 are available only when 99 tools can be registered.

(iii) Special tool

```
 101  #L    OUTER       T0101   RN     0. 400  AC      95  AN      80  TW     20. 000
 102  #R    OUTER       T0202   RN     0. 400  AC      95  AN      55  TW     20. 000
 103  #L    OUTER       T0303   RN     0. 400  AC      95  AN      55  TW     20. 000
 251  #L    OUTER  TH   T0404   RN     0. 400  AN      60              TW     20. 000
 252  #R    OUTER  TH   T0505   RN     0. 400  AN      60              TW     20. 000

*** MACHINING CONDITION (SIDECUT) ***
[#9990=x]                             ROUGHING     [#9990=y]                     FINISHING
SPECIAL SIDECUT (S-WDTH MM~           MM)          SPECIAL SIDECUT
MATERIAL        FEED/REV.  CUT SPD.                MATERIAL       FEED/REV.  CUT SPD.
                (MM/EDG.)  (M/MIN.)                               (MM/EDG.)  (M/MIN.)
<#9991=1>       #9992      #9993                   <#9991=1>      #9992      #9993
<#9991=2>       #9992      #9993                   <#9991=2>      #9992      #9993
<#9991=3>       #9992      #9993                   <#9991=3>      #9992      #9993
<#9991=4>       #9992      #9993                   <#9991=4>      #9992      #9993
<#9991=5>       #9992      #9993                   <#9991=5>      #9992      #9993
<#9991=6>       #9992      #9993                   <#9991=6>      #9992      #9993
<#9991=7>       #9992      #9993                   <#9991=7>      #9992      #9993
<#9991=8>       #9992      #9993                   <#9991=8>      #9992      #9993
                                                   SET FEEDRATE AMOUNT BY MAX. VALUE.

|  WORK  | ENDMIL | SIDCUT | CHAMFR |  |     |  |  |  MENU  |  |  |  |
| MATERL |        |        |        |  |     |  |  |  RETRN |  |  |  |
```

- Workpiece material 1 to 8 : x=9011, y=9012
- Workpiece material 9 to 16 : x=9071, y=9072
- Workpiece material 17 to 24 : x=9077, y=9078

**NOTE**
Workpiece materials 9 to 16 and 17 to 24 are available only when 99 tools can be registered.

(c) Chamfering tool

(i) High speed tool

```
 101  #L    OUTER        T0101   RN       0. 400  AC       95  AN       80  TW       20. 000
 102  #R    OUTER        T0202   RN       0. 400  AC       95  AN       55  TW       20. 000
 103  #L    OUTER        T0303   RN       0. 400  AC       95  AN       55  TW       20. 000
 251  #L    OUTER  TH    T0404   RN       0. 400  AN       60                   TW       20. 000
 252  #R    OUTER  TH    T0505   RN       0. 400  AN       60                   TW       20. 000

*** MACHINING CONDITION (CHAMFER) ***
[#9990=x]
HI-SPD CHAMFR     (S-WDTH   MM~       MM)
MATERIAL          FEED/REV. CUT SPD.
                  (MM/EDG.) (M/MIN.)
<#9991=1>         #9992     #9993
<#9991=2>         #9992     #9993
<#9991=3>         #9992     #9993
<#9991=4>         #9992     #9993
<#9991=5>         #9992     #9993
<#9991=6>         #9992     #9993
<#9991=7>         #9992     #9993
<#9991=8>         #9992     #9993

                                      SET FEEDRATE AMOUNT BY MAX. VALUE.

| WORK   | ENDMIL | SIDCUT | CHAMFR |  |  |  | MENU  |  |  |
| MATERL |        |        |        |  |  |  | RETRN |  |  |
```

- Workpiece material 1 to 8 : x=9014
- Workpiece material 9 to 16 : x=9080
- Workpiece material 17 to 24 : x=9083

**NOTE**

Workpiece materials 9 to 16 and 17 to 24 are available only when 99 tools can be registered.

(ii) Carbide tool

```
 101  #L   OUTER      T0101  RN   0. 400  AC   95  AN   80  TW   20. 000
 102  #R   OUTER      T0202  RN   0. 400  AC   95  AN   55  TW   20. 000
 103  #L   OUTER      T0303  RN   0. 400  AC   95  AN   55  TW   20. 000
 251  #L   OUTER  TH  T0404  RN   0. 400  AN   60                TW   20. 000
 252  #R   OUTER  TH  T0505  RN   0. 400  AN   60                TW   20. 000

*** MACHINING CONDITION (CHAMFER) ***
[#9990=x]
CARBID CHAMFR (S-WDTH   MM~       MM)
MATERIAL      FEED/REV. CUT SPD.
              (MM/EDG.) (M/MIN.)
<#9991=1>     #9992     #9993
<#9991=2>     #9992     #9993
<#9991=3>     #9992     #9993
<#9991=4>     #9992     #9993
<#9991=5>     #9992     #9993
<#9991=6>     #9992     #9993
<#9991=7>     #9992     #9993
<#9991=8>     #9992     #9993
                                SET FEEDRATE AMOUNT BY MAX. VALUE.

|  | WORK   | ENDMIL | SIDCUT | CHAMFR |  |  |  |  | MENU  |  |  |  |
|  | MATERL |        |        |        |  |  |  |  | RETRN |  |  |  |
```

- Workpiece material 1 to 8 : x=9013
- Workpiece material 9 to 16 : x=9079
- Workpiece material 17 to 24 : x=9082

> **NOTE**
> Workpiece materials 9 to 16 and 17 to 24 are available only when 99 tools can be registered.

(iii) Special tool

```
 101  #L    OUTER       T0101  RN    0. 400  AC     95  AN     80  TW    20. 000
 102  #R    OUTER       T0202  RN    0. 400  AC     95  AN     55  TW    20. 000
 103  #L    OUTER       T0303  RN    0. 400  AC     95  AN     55  TW    20. 000
 251  #L    OUTER  TH   T0404  RN    0. 400  AN     60                TW    20. 000
 252  #R    OUTER  TH   T0505  RN    0. 400  AN     60                TW    20. 000

*** MACHINING CONDITION (CHAMFER) ***
[#9990=x]
SPECIAL CHAMFR(S-WDTH   MM~       MM)
MATERIAL      FEED/REV. CUT SPD.
              (MM/EDG.) (M/MIN.)
<#9991=1>     #9992     #9993
<#9991=2>     #9992     #9993
<#9991=3>     #9992     #9993
<#9991=4>     #9992     #9993
<#9991=5>     #9992     #9993
<#9991=6>     #9992     #9993
<#9991=7>     #9992     #9993
<#9991=8>     #9992     #9993
                                 SET FEEDRATE AMOUNT BY MAX. VALUE.

|  | WORK MATERL | ENDMIL | SIDCUT | CHAMFR |  |  |  | MENU RETRN |  |  |  |
```

- Workpiece material 1 to 8 : x=9015
- Workpiece material 9 to 16 : x=9081
- Workpiece material 17 to 24 : x=9084

**NOTE**

Workpiece materials 9 to 16 and 17 to 24 are available only when 99 tools can be registered.

(7) Coefficient setting data screen

(a) Coefficient setting data for general-purpose tools

```
101  #L   OUTER     T0101  RN     0. 400  AC      95  AN      80  TW     20. 000
102  #R   OUTER     T0202  RN     0. 400  AC      95  AN      55  TW     20. 000
103  #L   OUTER     T0303  RN     0. 400  AC      95  AN      55  TW     20. 000
251  #L   OUTER  TH T0404  RN     0. 400  AN      60                  TW     20. 000
252  #R   OUTER  TH T0505  RN     0. 400  AN      60                  TW     20. 000

***  MACHINING  CONDITION  (COEFIC)  ***
[#9990=x]    FINISHING         GENERAL
FEED/REV.
(MM/REV.)    #9992 #9993 #9994 #9995 <#9991=1>
MATERIAL
<#9991=2> #9992 #9993 #9994 #9995 #9996
<#9991=3> #9992 #9993 #9994 #9995 #9996
<#9991=4> #9992 #9993 #9994 #9995 #9996
<#9991=5> #9992 #9993 #9994 #9995 #9996
<#9991=6> #9992 #9993 #9994 #9995 #9996
<#9991=7> #9992 #9993 #9994 #9995 #9996
<#9991=8> #9992 #9993 #9994 #9995 #9996
<#9991=9> #9992 #9993 #9994 #9995 #9996
                                   (CUT SPD.)

| WORK MATERL |  |  |  |  |  |  | MENU RETRN |  | GUIDE |
```

- Workpiece material 1 to 8 : x= 31
- Workpiece material 9 to 16 : x=203
- Workpiece material 17 to 24 : x=204

**NOTE**

Workpiece materials 9 to 16 and 17 to 24 are available only when 99 tools can be registered.

(b) Coefficient setting data for threading tools

```
 101  #L    OUTER       T0101  RN    0. 400  AC    95  AN    80  TW    20. 000
 102  #R    OUTER       T0202  RN    0. 400  AC    95  AN    55  TW    20. 000
 103  #L    OUTER       T0303  RN    0. 400  AC    95  AN    55  TW    20. 000
 251  #L    OUTER  TH   T0404  RN    0. 400  AN    60                TW    20. 000
 252  #R    OUTER  TH   T0505  RN    0. 400  AN    60                TW    20. 000

 ***  MACHINING  CONDITION  (COEFIC)  ***
 [#9990=x]                             THREAD
 LEAD
 (MM/REV.)    #9992 #9993 #9994 #9995<#9991=1>
 MATERIAL
 <#9991=2> #9992 #9993 #9994 #9995 #9996
 <#9991=3> #9992 #9993 #9994 #9995 #9996
 <#9991=4> #9992 #9993 #9994 #9995 #9996
 <#9991=5> #9992 #9993 #9994 #9995 #9996
 <#9991=6> #9992 #9993 #9994 #9995 #9996
 <#9991=7> #9992 #9993 #9994 #9995 #9996
 <#9991=8> #9992 #9993 #9994 #9995 #9996
 <#9991=9> #9992 #9993 #9994 #9995 #9996
                                   (CUT DPTH)

 | WORK   |  |  |  |  |  |  | MENU  |  | GUIDE |
 | MATERL |  |  |  |  |  |  | RETRN |  |       |
```

- Workpiece material 1 to 8 : x= 99
- Workpiece material 9 to 16 : x=205
- Workpiece material 17 to 24 : x=206

**NOTE**

Workpiece materials 9 to 16 and 17 to 24 are available only when 99 tools can be registered.

(c) Coefficient setting data for grooving tools

```
 101  #L    OUTER      T0101  RN     0. 400  AC       95  AN       80  TW      20. 000
 102  #R    OUTER      T0202  RN     0. 400  AC       95  AN       55  TW      20. 000
 103  #L    OUTER      T0303  RN     0. 400  AC       95  AN       55  TW      20. 000
 251  #L    OUTER  TH  T0404  RN     0. 400  AN       60                TW      20. 000
 252  #R    OUTER  TH  T0505  RN     0. 400  AN       60                TW      20. 000

 ***  MACHINING  CONDITION  (COEFIC)  ***
[#9990=32]                              GROOVE

GRV  WIDTH   #9992 #9993 #9994 #9995<#9991=1>
  (MM)
 CARBID   #9992 #9993 #9994 #9995 #9996<#9991=2>
 HI-SPD   #9992 #9993 #9994 #9995 #9996<#9991=3>
 SPECIAL  #9992 #9993 #9994 #9995 #9996<#9991=4>

                                      (FEED/REV.)

                                        MENU            GUIDE
                                        RETRN
```

(d) Coefficient setting data for drills

```
 101  #L   OUTER        T0101  RN    0. 400  AC     95  AN     80  TW     20. 000
 102  #R   OUTER        T0202  RN    0. 400  AC     95  AN     55  TW     20. 000
 103  #L   OUTER        T0303  RN    0. 400  AC     95  AN     55  TW     20. 000
 251  #L   OUTER   TH   T0404  RN    0. 400  AN     60                 TW     20. 000
 252  #R   OUTER   TH   T0505  RN    0. 400  AN     60                 TW     20. 000

 ***  MACHINING  CONDITION  (COEFIC)  ***
 [#9990=33]                                   DRIL

 NOMINL-D      #9992 #9993 #9994 #9995<#9991=1>
   (MM)
  HI-SPD   #9992 #9993 #9994 #9995 #9996<#9991=3>
  CARBID   #9992 #9993 #9994 #9995 #9996<#9991=2>
  SPECIAL  #9992 #9993 #9994 #9995 #9996<#9991=4>

                                   (FEED/REV.)

 |    |    |    |    |    |    |    |    | MENU  |    | GUIDE |    |
                                           RETRN
```

(e) Coefficient setting data for center drilling tools

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 101 | #L | OUTER | | T0101 | RN | 0. 400 | AC | 95 | AN | 80 | TW | 20. 000 |
| 102 | #R | OUTER | | T0202 | RN | 0. 400 | AC | 95 | AN | 55 | TW | 20. 000 |
| 103 | #L | OUTER | | T0303 | RN | 0. 400 | AC | 95 | AN | 55 | TW | 20. 000 |
| 251 | #L | OUTER | TH | T0404 | RN | 0. 400 | AN | 60 | | | TW | 20. 000 |
| 252 | #R | OUTER | TH | T0505 | RN | 0. 400 | AN | 60 | | | TW | 20. 000 |

```
*** MACHINING CONDITION (COEFIC) ***
[#9990=95]                            CENTER

NOMINL-D      #9992 #9993 #9994 #9995<#9991=1>
  (MM)
 HI-SPD   #9992 #9993 #9994 #9995 #9996<#9991=3>
 CARBID   #9992 #9993 #9994 #9995 #9996<#9991=2>
 SPECIAL  #9992 #9993 #9994 #9995 #9996<#9991=4>

                                  (FEED/REV.)
```

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | MENU RETRN | | GUIDE | |

(f) Coefficient setting data for reamers

```
 101  #L    OUTER        T0101  RN     0. 400  AC      95  AN      80  TW      20. 000
 102  #R    OUTER        T0202  RN     0. 400  AC      95  AN      55  TW      20. 000
 103  #L    OUTER        T0303  RN     0. 400  AC      95  AN      55  TW      20. 000
 251  #L    OUTER  TH    T0404  RN     0. 400  AN      60              TW      20. 000
 252  #R    OUTER  TH    T0505  RN     0. 400  AN      60              TW      20. 000

***  MACHINING  CONDITION  (COEFIC)  ***
[#9990=207]                                  REAMER

BORE DIA.     #9992 #9993 #9994 #9995<#9991=1>
  (MM)
 HI-SPD   #9992 #9993 #9994 #9995 #9996<#9991=3>
 CARBID   #9992 #9993 #9994 #9995 #9996<#9991=2>
 SPECIAL  #9992 #9993 #9994 #9995 #9996<#9991=4>

                                     (FEED/REV.)

|    |    |    |    |    |    |    |    | MENU  |    | GUIDE |    |
|    |    |    |    |    |    |    |    | RETRN |    |       |    |
```

(g) Coefficient setting data for boring tools

```
 101  #L   OUTER       T0101  RN    0. 400  AC     95  AN     80  TW     20. 000
 102  #R   OUTER       T0202  RN    0. 400  AC     95  AN     55  TW     20. 000
 103  #L   OUTER       T0303  RN    0. 400  AC     95  AN     55  TW     20. 000
 251  #L   OUTER  TH   T0404  RN    0. 400  AN     60                TW     20. 000
 252  #R   OUTER  TH   T0505  RN    0. 400  AN     60                TW     20. 000

*** MACHINING CONDITION (COEFIC) ***
[#9990=208]                                BORE

BORE DIA.    #9992 #9993 #9994 #9995<#9991=1>
  (MM)
 CARBID   #9992 #9993 #9994 #9995 #9996<#9991=2>
 HI-SPD   #9992 #9993 #9994 #9995 #9996<#9991=3>
 SPECIAL  #9992 #9993 #9994 #9995 #9996<#9991=4>




                                    (FEED/REV.)

|    |    |    |    |    |    |    |    | MENU  |    | GUIDE |    |
|    |    |    |    |    |    |    |    | RETRN |    |       |    |
```

(h) Coefficient setting data for taps

```
 101  #L    OUTER       T0101  RN     0. 400  AC      95  AN      80  TW      20. 000
 102  #R    OUTER       T0202  RN     0. 400  AC      95  AN      55  TW      20. 000
 103  #L    OUTER       T0303  RN     0. 400  AC      95  AN      55  TW      20. 000
 251  #L    OUTER  TH   T0404  RN     0. 400  AN      60                  TW      20. 000
 252  #R    OUTER  TH   T0505  RN     0. 400  AN      60                  TW      20. 000

*** MACHINING  CONDITION  (COEFIC)  ***
[#9990=34]                                   TAP

NOMINL-D      #9992 #9993 #9994 #9995<#9991=1>
  (MM)
 HI-SPD   #9992 #9993 #9994 #9995 #9996<#9991=3>
 CARBID   #9992 #9993 #9994 #9995 #9996<#9991=2>
 SPECIAL  #9992 #9993 #9994 #9995 #9996<#9991=4>

                                  (CUT SPD.)

                                          MENU         GUIDE
                                          RETRN
```

(i) Coefficient setting data for end mills

```
101  #L   OUTER      T0101  RN    0. 400  AC     95  AN    80  TW    20. 000
102  #R   OUTER      T0202  RN    0. 400  AC     95  AN    55  TW    20. 000
103  #L   OUTER      T0303  RN    0. 400  AC     95  AN    55  TW    20. 000
251  #L   OUTER  TH  T0404  RN    0. 400  AN     60              TW    20. 000
252  #R   OUTER  TH  T0505  RN    0. 400  AN     60              TW    20. 000

***  MACHINING  CONDITION  (COEFIC)  ***
[#9990=9016]       ROUGHING          ENDMIL
NOMINL-D     #9992 #9993 #9994 #9995<#9991=1>
  (MM)
 CARBID   #9992 #9993 #9994 #9995 #9996<#9991=2>
 HI-SPD   #9992 #9993 #9994 #9995 #9996<#9991=3>
 SPECIAL  #9992 #9993 #9994 #9995 #9996<#9991=4>

[#9990=9017]       FINISHING
NOMINL-D     #9992 #9993 #9994 #9995<#9991=1>
  (MM)
 CARBID   #9992 #9993 #9994 #9995 #9996<#9991=2>
 HI-SPD   #9992 #9993 #9994 #9995 #9996<#9991=3>
 SPECIAL  #9992 #9993 #9994 #9995 #9996<#9991=4>
                                  (FEED/REV.)

                                     MENU            GUIDE
                                     RETRN
```

(j) Coefficient setting data for side cutters

```
 101  #L    OUTER       T0101   RN     0. 400  AC       95  AN       80  TW      20. 000
 102  #R    OUTER       T0202   RN     0. 400  AC       95  AN       55  TW      20. 000
 103  #L    OUTER       T0303   RN     0. 400  AC       95  AN       55  TW      20. 000
 251  #L    OUTER  TH   T0404   RN     0. 400  AN       60                TW      20. 000
 252  #R    OUTER  TH   T0505   RN     0. 400  AN       60                TW      20. 000

 ***  MACHINING  CONDITION  (COEFIC)  ***
 [#9990=9018]       ROUGHING          SIDECUT
 NOMINL-D     #9992 #9993 #9994 #9995<#9991=1>
   (MM)
  CARBID   #9992 #9993 #9994 #9995 #9996<#9991=2>
  HI-SPD   #9992 #9993 #9994 #9995 #9996<#9991=3>
  SPECIAL  #9992 #9993 #9994 #9995 #9996<#9991=4>

 [#9990=9019]       FINISHING
 NOMINL-D     #9992 #9993 #9994 #9995<#9991=1>
   (MM)
  CARBID   #9992 #9993 #9994 #9995 #9996<#9991=2>
  HI-SPD   #9992 #9993 #9994 #9995 #9996<#9991=3>
  SPECIAL  #9992 #9993 #9994 #9995 #9996<#9991=4>
                                   (FEED/REV.)

                                          MENU            GUIDE
                                          RETRN
```

(8) Surface roughness data screen

```
 101  #L    OUTER       T0101   RN     0. 400  AC       95  AN       80  TW      20. 000
 102  #R    OUTER       T0202   RN     0. 400  AC       95  AN       55  TW      20. 000
 103  #L    OUTER       T0303   RN     0. 400  AC       95  AN       55  TW      20. 000
 251  #L    OUTER  TH   T0404   RN     0. 400  AN       60                TW      20. 000
 252  #R    OUTER  TH   T0505   RN     0. 400  AN       60                TW      20. 000

*** MACHINING  CONDITION  (ROUGH)  ***
[#9990=35]
RGH CODE          SURFACE ROUGHNESS
                  (MM/100)
   1 ▽            #9992<#9991=1>
   2 ▽            #9992<#9991=2>
   3 ▽▽           #9992<#9991=3>
   4 ▽▽           #9992<#9991=4>
   5 ▽▽▽          #9992<#9991=5>
   6 ▽▽▽          #9992<#9991=6>
   7 ▽▽▽          #9992<#9991=7>
   8 ▽▽▽          #9992<#9991=8>
   9 ▽▽▽          #9992<#9991=9>
  10 ▽▽▽          #9992<#9991=10>

                                     MENU
                                     RETRN
```

(9) Pre-tool list

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 101 | #L | OUTER | | T0101 | RN | 0. 400 | AC | 95 | AN | 80 | TW | 20. 000 |
| 102 | #R | OUTER | | T0202 | RN | 0. 400 | AC | 95 | AN | 55 | TW | 20. 000 |
| 103 | #L | OUTER | | T0303 | RN | 0. 400 | AC | 95 | AN | 55 | TW | 20. 000 |
| 251 | #L | OUTER | TH | T0404 | RN | 0. 400 | AN | 60 | | | TW | 20. 000 |
| 252 | #R | OUTER | TH | T0505 | RN | 0. 400 | AN | 60 | | | TW | 20. 000 |

```
***  PRE-TOOL  LIST  ***
[#9990=x]
                TOOL NO.             TL NAME
LAST  TL        #9992<#9991=1>

PRE-TOOL  1     #9992<#9991=2>
          2     #9992<#9991=3>
          3     #9992<#9991=4>
          4     #9992<#9991=5>
          5     #9992<#9991=6>
```

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | WORK MATERL | | | | | SEARCH | | MENU RETRN | TOOL LIST | | |

- Pre-tool list 1 : x=101
- Pre-tool list 2 : x=102
- Pre-tool list 3 : x=103
- Pre-tool list 4 : x=104
- Pre-tool list 5 : x=105
- Pre-tool list 6 : x=106

(10) Chuck/tailstock figure data

(a) Chuck figure data

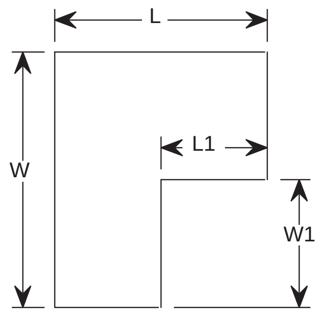
To read the data for a chuck, using system variables #9951 to #9955, set the corresponding chuck number in #9950.

```
101  #L    OUTER       T0101  RN     0. 400  AC      95  AN      80  TW      20. 000
102  #R    OUTER       T0202  RN     0. 400  AC      95  AN      55  TW      20. 000
103  #L    OUTER       T0303  RN     0. 400  AC      95  AN      55  TW      20. 000
251  #L    OUTER  TH   T0404  RN     0. 400  AN      60                  TW      20. 000
252  #R    OUTER  TH   T0505  RN     0. 400  AN      60                  TW      20. 000

***  CHUCK  FIGURE  ***

NO.  TYPE   L      W      L1     W1
 1   #9951  #9952  #9953  #9954  #9955 [#9950= 1]
 2   #9951  #9952  #9953  #9954  #9955 [#9950= 2]
 3   #9951  #9952  #9953  #9954  #9955 [#9950= 3]
 4   #9951  #9952  #9953  #9954  #9955 [#9950= 4]
 5   #9951  #9952  #9953  #9954  #9955 [#9950= 5]
 6   #9951  #9952  #9953  #9954  #9955 [#9950= 6]
 7   #9951  #9952  #9953  #9954  #9955 [#9950= 7]
 8   #9951  #9952  #9953  #9954  #9955 [#9950= 8]
 9   #9951  #9952  #9953  #9954  #9955 [#9950= 9]
10   #9951  #9952  #9953  #9954  #9955 [#9950=10]

     EXT.   INT.   OUT-S   CHUCK   TAIL            MENU
                                   STOCK           RETRN
```




<TYPE>

External jaw : #9951=1
Internal jaw : #9551=2
Special external jaw : #9551=3

> **NOTE**
> The OUT-S type is supported only when the optional 60-chuck function is provided. This function provides data on 60 different chucks.

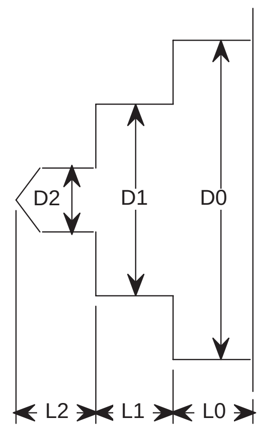
(b) Tailstock figure data

To read the data for a tailstock, using system variables #9981 to #9986, set the desired tailstock number in #9980.

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 101 | #L | OUTER | | T0101 | RN | 0. 400 | AC | 95 | AN | 80 | TW | 20. 000 |
| 102 | #R | OUTER | | T0202 | RN | 0. 400 | AC | 95 | AN | 55 | TW | 20. 000 |
| 103 | #L | OUTER | | T0303 | RN | 0. 400 | AC | 95 | AN | 55 | TW | 20. 000 |
| 251 | #L | OUTER | TH | T0404 | RN | 0. 400 | AN | 60 | | | TW | 20. 000 |
| 252 | #R | OUTER | TH | T0505 | RN | 0. 400 | AN | 60 | | | TW | 20. 000 |

*** TAIL STOCK FIGURE ***

| NO. | 1<br>[#9980=1] | 2<br>[#9980=1] |
|---|---|---|
| D0 | #9981 | #9981 |
| L0 | #9982 | #9982 |
| D1 | #9983 | #9983 |
| L1 | #9984 | #9984 |
| D2 | #9985 | #9985 |
| L2 | #9986 | #9986 |




| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | CHUCK | TAIL STOCK | | | MENU RETRN | | |

**NOTE**

Data for up to four tailstocks can be stored.

## E.6.2 Software Package Associated with Process Data

### E.6.2.1 Overview

With the FANUC Super CAP T Ver. 1, 2, 3 the first macro variable number of the process data of a process in a conversational program can be read into #9898 to reference the process data. In addition, a process can be searched for using the process data.

### E.6.2.2 Specifications

The first macro variable number of a process to be searched for can be set in #9898 by setting reference process data used for search operation in #9892, setting the offset number of the macro variable containing data to be searched for in #9893, and setting the completion code in #9899. If search operation fails, the error code (99) is set in #9898.

In #9890, the first macro variable number of the process data of the currently indicated process is set. In addition, the initially set first macro variable number of the program currently selected is set in #9891.

(a) Macro variables and process numbers used with the function

(i) Details of macro variables

#9898 – When a process is searched for using process data, the first macro variable number of the process searched for is set. (This macro variable allows read operation only.)

#9899 – In process search operation, –1 is to be entered when reference process data for search operation and the offset number, from the beginning, of the macro variable containing data to be searched have been entered. (This macro variable allows write operation only.)

#9890 – The first macro variable number of the process data currently edited is set. (This macro variable allows read operation only.)

#9891 – The first macro variable number of the initially set data of the program currently edited is set. (This macro variable allows read operation only.)

#9880 – The first macro variable number of the process data currently executed is set. (This macro variable allows read operation only.)

#9881 – The first macro variable number of the initially set data of the program currently executed is set. (This macro variable allows read operation only.)

#9884 – The first macro variable number of the block of the initially set workpiece figure data of the current program is set. (This macro variable allows read operations only.)

#9892 – When a process is to be searched for, reference process data used for search operation is set. (This macro variable allows write operation only.)

#9893 – When a process is to be searched for, the offset number, from the beginning, of the macro variable containing data to be searched for is set. (This macro variable allows write operation only.)

#9894 – The first macro variable number of the block containing the initially set molding material figure data of the program currently edited is set. (This macro variable allows read operation only.)

#9895 = 1: Searches for the program currently executed.
0: Searches for the program currently edited.

P-code variables #10000 to #10699 can be freely used with a user program. With a general macro compiler/executor (with the FANUC Super CAP T Ver. 1 not incorporated), the number of P-code variables (#10000 and up) can be changed using compile parameter number 9037. With the system that has the conversational function, however, this parameter is always set to 7. The system uses extended P-code variables #20000 and up; these variables cannot be used with a user program.

(ii) Process numbers

Currently, up to 99 processes can be registered. When processes are generated, the same number is assigned to a rough machining process, finish machining process, and chamfering process. (After renumbering operation is executed, a different process number is assigned in ascending order to each of these processes.)

When a workpiece figure represents a molding material figure, 60 macro variables (initial setting - 2) are added to contain the data of each point of the workpiece figure at the time of initial setting. The first macro variable number of the block containing the molding material figure data is set in #9894. When a workpiece figure does not represent a molding material figure, #9894 is set to 0 or a null.

(b) Process data and offset

A conversational program consists of a block of 60 macro variables as a basic unit. So, except at the time of initial setting, any process data can be represented using the first macro variable number of the block containing the process and its offset from the first macro variable number.

However, the contents of a program depends on the type of process. Accordingly, the data represented by the same offset can vary from process to process.

(i) Details of offsets for each type of process

<Initial Setting – 1>

| | | |
|---|---|---|
| + 0 | Work material | |
| + 1 | Work figure (1=Bar, 2=Molding) | |
| + 2 | Outside diameter D (Bar) | |
| + 3 | Inside diameter H (Bar) | |
| + 4 | Length L (Bar) | |
| + 5 | Unused | |
| + 6 | Unused | |
| + 7 | T code T | (2) |
| + 8 | Workpiece shift amount SZ | (2) |
| + 9 | Chuck number C | (2) |
| +10 | Chuck reference point X | (2) |
| +11 | Chuck reference point Z | (2) |
| +12 | Product length PL | |
| +13 | Coolant (1=ON, 2=OFF) | (∗1) |
| +14 | ●Trapezoidal groove figure editing flag | (∗22) |
| +15 | ●Tail stock number | (2) |
| +16 | ●Tool change position X | (1) |
| +17 | ●Tool change position Z | (1) |
| +18 | ●Tool change position X | (2) |
| +19 | ●Tool change position Z | (2) |
| +20 | ●Tail stock reference point Z | (1) |
| +21 | ●Tail stock reference point Z | (2) |
| +22 | ●Face position | |
| +23 | Program prepared data (y, m, d) | |
| +24 | Program prepared data (time) | |
| +25 | Program update (y, m, d) | |
| +26 | Program update (time) | |
| +27 | Common safety point X for drilling (turning) | |
| +28 | Common safety point Z for drilling (turning) | |
| +29 | Maximum spindle speed | |

| | | |
|---|---|---|
| +30 | Finishing allowance X | (∗2) |
| +31 | Finishing allowance Z | (∗2) |
| +32 | End face cutting allowance E, CZ | |
| +33 | T code T | (1) |
| +34 | Workpiece shift amount SZ | (1) |
| +35 | Chuck number C | (1) |
| +36 | Chuck reference point X | (1) |
| +37 | Chuck reference point Z | (1) |
| +38 | Tail stock number | |
| +39 | | |
| +40 | | |
| +41 | Name of program ASCII code | |
| +42 | | |
| +43 | | |
| +44 | Unused | |
| +45 | Unused | |
| +46 | Unused | |
| +47 | System utilization area | (∗3) |
| +48 | Run hour | |
| +49 | Common safety point outer diameter X | |
| +50 | Common safety point outer diameter Z | |
| +51 | Common safety point internal diameter X | |
| +52 | Common safety point internal diameter Z | |
| +53 | Unused | |
| +54 | Unused | |
| +55 | First variable number of the 2nd initial setting block | |
| +56 | * Unused | |
| +57 | * Use status flag (0: Not used, 1: Used) | |
| +58 | * Unused | |
| +59 | * First variable number of the next block | |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

> **NOTE**
> Some item names vary with the version.

[1]: First tool post [2]: Second tool post (∗?): See the note below.

* : System management area –
Avoid data writing by a user program.

<Initial Setting – 2>

| | | | | | |
|---|---|---|---|---|---|
| + 0 | Outside diameter X1 | (Molding) | +30 | Inside diameter X4 | (Molding) |
| + 1 | Outside diameter Z1 | (Molding) | +31 | Inside diameter Z4 | (Molding) |
| + 2 | Outside diameter X2 | (Molding) | +32 | Inside diameter X5 | (Molding) |
| + 3 | Outside diameter Z2 | (Molding) | +33 | Inside diameter Z5 | (Molding) |
| + 4 | Outside diameter X3 | (Molding) | +34 | Inside diameter X6 | (Molding) |
| + 5 | Outside diameter Z3 | (Molding) | +35 | Inside diameter Z6 | (Molding) |
| + 6 | Outside diameter X4 | (Molding) | +36 | Inside diameter X7 | (Molding) |
| + 7 | Outside diameter Z4 | (Molding) | +37 | Inside diameter Z7 | (Molding) |
| + 8 | Outside diameter X5 | (Molding) | +38 | Inside diameter X8 | (Molding) |
| + 9 | Outside diameter Z5 | (Molding) | +39 | Inside diameter Z8 | (Molding) |
| +10 | Outside diameter X6 | (Molding) | +40 | Inside diameter X9 | (Molding) |
| +11 | Outside diameter Z6 | (Molding) | +41 | Inside diameter Z9 | (Molding) |
| +12 | Outside diameter X7 | (Molding) | +42 | Inside diameter X10 | (Molding) |
| +13 | Outside diameter Z7 | (Molding) | +43 | Inside diameter Z10 | (Molding) |
| +14 | Outside diameter X8 | (Molding) | +44 | Inside diameter X11 | (Molding) |
| +15 | Outside diameter Z8 | (Molding) | +45 | Inside diameter Z11 | (Molding) |
| +16 | Outside diameter X9 | (Molding) | +46 | Inside diameter X12 | (Molding) |
| +17 | Outside diameter Z9 | (Molding) | +47 | Inside diameter Z12 | (Molding) |
| +18 | Outside diameter X10 | (Molding) | +48 | Unused | |
| +19 | Outside diameter Z10 | (Molding) | +49 | Unused | |
| +20 | Outside diameter X11 | (Molding) | +50 | Unused | |
| +21 | Outside diameter Z11 | (Molding) | +51 | Unused | |
| +22 | Outside diameter X12 | (Molding) | +52 | Unused | |
| +23 | Outside diameter Z12 | (Molding) | +53 | Unused | |
| +24 | Inside diameter X1 | (Molding) | +54 | Unused | |
| +25 | Inside diameter Z1 | (Molding) | +55 | * Unused | |
| +26 | Inside diameter X2 | (Molding) | +56 | * Unused | |
| +27 | Inside diameter Z2 | (Molding) | +57 | * Use status flag (0: Not used, 1: Used) | |
| +28 | Inside diameter X3 | (Molding) | +58 | * Unused | |
| +29 | Inside diameter Z3 | (Molding) | +59 | * Unused | |

●: Ver.2 or later, ■: Ver.3 or later.

**NOTE**
Some item names vary with the version.

* : System management area –
Avoid data writing by a user program.

<Process Data>

- Bar (rough machining)

| | | | |
|---|---|---|---|
| + 0 | Process number | +30 | Speed <residual machining> |
| + 1 | Unused | +31 | Surface reoughness <residual machining> |
| + 2 | System utilization area (∗3) | +32 | ●Escape amount <residual machining> |
| + 3 | Machining type (∗4) | +33 | ●Process movement (1=Standard, 2=High speed)<residual machining> |
| + 4 | Machining area (∗5) | +34 | Tool number <residual machining> |
| + 5 | Tool post (Spindle axis selection) (∗6) | +35 | T code <residual machining> |
| + 6 | Machining cycle (∗7) | +36 | ●Program override |
| + 7 | Cutting start point X | +37 | ●Program override <residual machining> |
| + 8 | Cutting start point Z | +38 | Cutting speed <residual machining> |
| + 9 | Unused | +39 | Feed amount <residual machining> |
| +10 | Unused | +40 | Direction of rotation (1=CW, 2=CCW) <residual machining> |
| +11 | Surface roughness | +41 | Cut depth <residual machining> |
| +12 | ●Escape amount | +42 | Spindle gear (∗8) <residual machining> |
| +13 | ●Process movement (1=Standard, 2=High speed) | +43 | Coolant (1=ON, 2=OFF)<residual machining> |
| +14 | Tool number | +44 | ●Pass point 1 X |
| +15 | T code | +45 | ●Pass point 1 Z |
| +16 | Unused | +46 | ●Pass point 2 X |
| +17 | Unused | +47 | ●Pass point 2 Z |
| +18 | Cutting speed | +48 | Run hour |
| +19 | Feed amount | +49 | System utilization area (∗3) |
| +20 | Direction of rotation (1=CW, 2=CCW) | +50 | Surface speed/speed selection (∗9) |
| +21 | Cut depth | +51 | Speed |
| +22 | Finishing allowance X | +52 | * First variable number for roughing |
| +23 | Finishing allowance Z | +53 | * First variable number for finishing |
| +24 | Spindle gear (∗8) | +54 | * First variable number for chamfering |
| +25 | Coolant (1=ON, 2=OFF) | +55 | * First variable number of a figure block |
| +26 | Automatic residual machining (1=Used, 2=Not used) | +56 | * Unused |
| +27 | Cutting start point X <residual machining> | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Cutting start point Z <residual machining> | +58 | * First variable number of the preceding process |
| +29 | Surface speed/speed selection (∗9) <residual machining> | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

**NOTE**
Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

- Bar (finishing)

| | | |
|---|---|---|
| + 0 | Process number | |
| + 1 | Unused | |
| + 2 | System utilization area | (*3) |
| + 3 | Machining type | (*4) |
| + 4 | Machining area | (*5) |
| + 5 | Tool post (Spindle axis selection) | (*6) |
| + 6 | Machining cycle | (*7) |
| + 7 | Cutting start point X | |
| + 8 | Cutting start point Z | |
| + 9 | Unused | |
| +10 | Unused | |
| +11 | Surface roughness | |
| +12 | ●Escape amount | |
| +13 | Unused | |
| +14 | Tool number | |
| +15 | T code | |
| +16 | Unused | |
| +17 | Unused | |
| +18 | Cutting speed | |
| +19 | Feed amount | |
| +20 | Direction of rotation (1=CW, 2=CCW) | |
| +21 | Unused | |
| +22 | Finishing allowance X | |
| +23 | Finishing allowance Z | |
| +24 | Spindle gear | (*8) |
| +25 | Coolant (1=ON, 2=OFF) | |
| +26 | Automatic residual machining (1=Used, 2=Unused) | |
| +27 | Cutting start point X <residual machining> | |
| +28 | Cutting start point Z <residual machining> | |
| +29 | Surface speed/speed selection <residual machining> | (*9) |

| | | |
|---|---|---|
| +30 | Speed <residual machining> | |
| +31 | Surface reoughness <residual machining> | |
| +32 | ●Escape amount <residual machining> | |
| +33 | Unused | |
| +34 | Tool number <residual machining> | |
| +35 | T code <residual machining> | |
| +36 | Unused | |
| +37 | Unused | |
| +38 | Cutting speed <residual machining> | |
| +39 | Feed amount <residual machining> | |
| +40 | Direction of rotation (1=CW, 2=CCW) <residual machining> | |
| +41 | Unused | |
| +42 | Spindle gear (*8) <residual machining> | |
| +43 | Coolant (1=ON, 2=OFF)<residual machining> | |
| +44 | ●Pass point 1 X | |
| +45 | ●Pass point 1 Z | |
| +46 | ●Pass point 2 X | |
| +47 | ●Pass point 2 Z | |
| +48 | Run hour | |
| +49 | System utilization area | (*3) |
| +50 | Surface speed/speed selection | (*9) |
| +51 | Speed | |
| +52 | * First variable number for roughing | |
| +53 | * First variable number for finishing | |
| +54 | * First variable number for chamfering | |
| +55 | * First variable number of a figure block | |
| +56 | * Unused | |
| +57 | * Use status flag (0: Not used, 1: Used) | |
| +58 | * First variable number of the preceding process | |
| +59 | * First variable number of the next process | |

●: Ver.2 or later, ■: Ver.3 or later, (*?): See the note below.

**NOTE**
Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0

● Pattern Repeating (rough machining)

| | | | |
|---|---|---|---|
| + 0 | Process number | +30 | Unused |
| + 1 | Unused | +31 | Unused |
| + 2 | System utilization area (∗3) | +32 | Unused |
| + 3 | Machining type (∗4) | +33 | Unused |
| + 4 | Machining area (∗5) | +34 | Unused |
| + 5 | Tool post (Spindle axis selection) (∗6) | +35 | Unused |
| + 6 | Machining cycle (∗7) | +36 | ●Program override |
| + 7 | Cutting start point X | +37 | Unused |
| + 8 | Cutting start point Z | +38 | Unused |
| + 9 | Unused | +39 | Unused |
| +10 | Unused | +40 | Unused |
| +11 | Surface roughness | +41 | Unused |
| +12 | Cutting allowance X | +42 | Unused |
| +13 | Cutting allowance Z | +43 | Unused |
| +14 | Tool number | +44 | ●Pass point 1 X |
| +15 | T code | +45 | ●Pass point 1 Z |
| +16 | Unused | +46 | ●Pass point 2 X |
| +17 | Unused | +47 | ●Pass point 2 Z |
| +18 | Cutting speed | +48 | Run hour |
| +19 | Feed amount | +49 | System utilization area (∗3) |
| +20 | Direction of rotation (1=CW, 2=CCW) | +50 | Surface speed/speed selection (∗9) |
| +21 | Cut depth | +51 | Speed |
| +22 | Finishing allowance X | +52 | * First variable number for roughing |
| +23 | Finishing allowance Z | +53 | * First variable number for finishing |
| +24 | Spindle gear (∗8) | +54 | * First variable number for chamfering |
| +25 | Coolant (1=ON, 2=OFF) | +55 | * First variable number of a figure block |
| +26 | Unused | +56 | * Unused |
| +27 | Unused | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Unused | +58 | * First variable number of the preceding process |
| +29 | Unused | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

> **NOTE**
> Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

- Pattern Repeating (finish machining)

| | | | |
|---|---|---|---|
| + 0 | Process number | +30 | Unused |
| + 1 | Unused | +31 | Unused |
| + 2 | System utilization area (∗3) | +32 | Unused |
| + 3 | Machining type (∗4) | +33 | Unused |
| + 4 | Machining area (∗5) | +34 | Unused |
| + 5 | Tool post (Spindle axis selection) (∗6) | +35 | Unused |
| + 6 | Machining cycle (∗7) | +36 | Unused |
| + 7 | Cutting start point X | +37 | Unused |
| + 8 | Cutting start point Z | +38 | Unused |
| + 9 | Unused | +39 | Unused |
| +10 | Unused | +40 | Unused |
| +11 | Surface roughness | +41 | Unused |
| +12 | Unused | +42 | Unused |
| +13 | Unused | +43 | Unused |
| +14 | Tool number | +44 | ●Pass point 1 X |
| +15 | T code | +45 | ●Pass point 1 Z |
| +16 | Unused | +46 | ●Pass point 2 X |
| +17 | Unused | +47 | ●Pass point 2 Z |
| +18 | Cutting speed | +48 | Run hour |
| +19 | Feed amount | +49 | System utilization area (∗3) |
| +20 | Direction of rotation (1=CW, 2=CCW) | +50 | Surface speed/speed selection (∗9) |
| +21 | Unused | +51 | Speed |
| +22 | Finishing allowance X | +52 | * First variable number for roughing |
| +23 | Finishing allowance Z | +53 | * First variable number for finishing |
| +24 | Spindle gear (∗8) | +54 | * First variable number for chamfering |
| +25 | Coolant (1=ON, 2=OFF) | +55 | * First variable number of a figure block |
| +26 | Unused | +56 | * Unused |
| +27 | Unused | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Unused | +58 | * First variable number of the preceding process |
| +29 | Unused | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

**NOTE**
Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

- Residual Machining (rough machining)

| | | | |
|---|---|---|---|
| + 0 | Process number | +30 | Unused |
| + 1 | Unused | +31 | Unused |
| + 2 | System utilization area (∗3) | +32 | Unused |
| + 3 | Machining type (∗4) | +33 | Unused |
| + 4 | Machining area (∗5) | +34 | Unused |
| + 5 | Tool post (Spindle axis selection) (∗6) | +35 | Unused |
| + 6 | Machining cycle (∗7) | +36 | ●Program override |
| + 7 | Machining start point X | +37 | Unused |
| + 8 | Machining start point Z | +38 | Unused |
| + 9 | Unused | +39 | Unused |
| +10 | Unused | +40 | Unused |
| +11 | Surface roughness | +41 | Unused |
| +12 | Unused | +42 | Unused |
| +13 | ●Process movement (1=Standard, 2=High speed) | +43 | Unused |
| +14 | Tool number | +44 | ●Pass point 1 X |
| +15 | T code | +45 | ●Pass point 1 Z |
| +16 | Unused | +46 | ●Pass point 2 X |
| +17 | Unused | +47 | ●Pass point 2 Z |
| +18 | Cutting speed | +48 | Run hour |
| +19 | Feed amount | +49 | Unused |
| +20 | Direction of rotation (1=CW, 2=CCW) | +50 | Surface speed/speed selection (∗9) |
| +21 | Cut depth | +51 | Speed |
| +22 | Finishing allowance X | +52 | * First variable number for roughing |
| +23 | Finishing allowance Z | +53 | * First variable number for finishing |
| +24 | Spindle gear (∗8) | +54 | * First variable number for chamfering |
| +25 | Coolant (1=ON, 2=OFF) | +55 | * First variable number of a figure block |
| +26 | Unused | +56 | * Unused |
| +27 | Unused | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Unused | +58 | * First variable number of the preceding process |
| +29 | Unused | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

> **NOTE**
> Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

- Residual Machining (finish machining)

| | | | |
|---|---|---|---|
| + 0 | Process number | +30 | Unused |
| + 1 | Unused | +31 | Unused |
| + 2 | System utilization area (∗3) | +32 | Unused |
| + 3 | Machining type (∗4) | +33 | Unused |
| + 4 | Machining area (∗5) | +34 | Unused |
| + 5 | Tool post (Spindle axis selection) (∗6) | +35 | Unused |
| + 6 | Machining cycle (∗7) | +36 | ●Program override |
| + 7 | Machining start point X | +37 | Unused |
| + 8 | Machining start point Z | +38 | Unused |
| + 9 | Unused | +39 | Unused |
| +10 | Unused | +40 | Unused |
| +11 | Surface roughness | +41 | Unused |
| +12 | Unused | +42 | Unused |
| +13 | Unused | +43 | Unused |
| +14 | Tool number | +44 | ●Pass point 1 X |
| +15 | T code | +45 | ●Pass point 1 Z |
| +16 | Unused | +46 | ●Pass point 2 X |
| +17 | Unused | +47 | ●Pass point 2 Z |
| +18 | Cutting feed | +48 | Run hour |
| +19 | Feed amount | +49 | Unused |
| +20 | Direction of rotation (1=CW, 2=CCW) | +50 | Surface speed/speed selection (∗9) |
| +21 | Unused | +51 | Speed |
| +22 | Finishing allowance X | +52 | * First variable number for roughing |
| +23 | Finishing allowance Z | +53 | * First variable number for finishing |
| +24 | Spindle gear (∗8) | +54 | * First variable number for chamfering |
| +25 | Coolant (1=ON, 2=OFF) | +55 | * First variable number of a figure block |
| +26 | Unused | +56 | * Unused |
| +27 | Unused | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Unused | +58 | * First variable number of the preceding process |
| +29 | Unused | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

> **NOTE**
> Some item names depend on the version.

* : System management area –
Avoid data writing by a user program. If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

- End Facing (rough machining)

| | | | |
|---|---|---|---|
| + 0 | Tool number | +30 | Unused |
| + 1 | Unused | +31 | Unused |
| + 2 | System utilization area (∗3) | +32 | Unused |
| + 3 | Machining type (∗4) | +33 | Unused |
| + 4 | Unused | +34 | Unused |
| + 5 | Tool post (Spindle axis selection) (∗6) | +35 | Unused |
| + 6 | Machining cycle (∗7) | +36 | ●Program override |
| + 7 | Machining start point X | +37 | Unused |
| + 8 | Machining start point Z | +38 | Unused |
| + 9 | End point | +39 | Unused |
| +10 | Unused | +40 | Unused |
| +11 | Surface roughness | +41 | Unused |
| +12 | Unused | +42 | Unused |
| +13 | Unused | +43 | Unused |
| +14 | Tool number | +44 | ●Pass point 1 X |
| +15 | T code | +45 | ●Pass point 1 Z |
| +16 | Unused | +46 | ●Pass point 2 X |
| +17 | Unused | +47 | ●Pass point 2 Z |
| +18 | Cutting speed | +48 | Run hour |
| +19 | Feed amount | +49 | Unused |
| +20 | Direction of rotation (1=CW, 2=CCW) | +50 | Surface speed/speed selection (∗9) |
| +21 | Cut depth | +51 | Speed |
| +22 | Unused | +52 | * First variable number for roughing |
| +23 | Finishing allowance Z | +53 | * First variable number for finishing |
| +24 | Spindle gear (∗8) | +54 | * First variable number for chamfering |
| +25 | Coolant (1=ON, 2=OFF) | +55 | * First variable number of a figure block |
| +26 | Unused | +56 | * Unused |
| +27 | Unused | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Unused | +58 | * First variable number of the preceding process |
| +29 | Unused | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

**NOTE**
Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

- End Facing (finish machining)

| | | | |
|---|---|---|---|
| + 0 | Process number | +30 | Unused |
| + 1 | Unused | +31 | Unused |
| + 2 | System utilization area (∗3) | +32 | Unused |
| + 3 | Machining type (∗4) | +33 | Unused |
| + 4 | Unused | +34 | Unused |
| + 5 | Tool post (∗6) | +35 | Unused |
| + 6 | Machining cycle (∗7) | +36 | ●Program override |
| + 7 | Machining start point X | +37 | Unused |
| + 8 | Machining start point Z | +38 | Unused |
| + 9 | End point | +39 | Unused |
| +10 | Unused | +40 | Unused |
| +11 | Surface roughness | +41 | Unused |
| +12 | Unused | +42 | Unused |
| +13 | Unused | +43 | Unused |
| +14 | Tool number | +44 | ●Pass point 1 X |
| +15 | T code | +45 | ●Pass point 1 Z |
| +16 | Unused | +46 | ●Pass point 2 X |
| +17 | Unused | +47 | ●Pass point 2 Z |
| +18 | Cutting speed | +48 | Run hour |
| +19 | Feed amount | +49 | Unused |
| +20 | Direction of rotation (1=CW, 2=CCW) | +50 | Surface speed/speed selection (∗9) |
| +21 | Unused | +51 | Speed |
| +22 | Unused | +52 | * First variable number for roughing |
| +23 | Finishing allowance X | +53 | * First variable number for finishing |
| +24 | Spindle gear (∗8) | +54 | * First variable number for chamfering |
| +25 | Coolant (1=ON, 2=OFF) | +55 | * First variable number of a figure block |
| +26 | Unused | +56 | * Unused |
| +27 | Unused | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Unused | +58 | * First variable number of the preceding process |
| +29 | Unused | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

**NOTE**
Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

- Threading

| | | |
|---|---|---|
| + 0 | Process number | |
| + 1 | Unused | |
| + 2 | System utilization area | (∗3) |
| + 3 | Machining type | (∗4) |
| + 4 | Machining area | (∗5) |
| + 5 | Tool post (Spindle axis selection) | (∗6) |
| + 6 | Machining cycle | (∗7) |
| + 7 | Machining start point X | |
| + 8 | Machining start point Z | |
| + 9 | Unused | |
| +10 | Unused | |
| +11 | Unused | |
| +12 | Unused | |
| +13 | Unused | |
| +14 | Tool number | |
| +15 | T code | |
| +16 | Unused | |
| +17 | Unused | |
| +18 | Cutting speed | |
| +19 | Unused | |
| +20 | Direction of rotation (1=CW, 2=CCW) | |
| +21 | Cut depth | |
| +22 | Unused | |
| +23 | Unused | |
| +24 | Spindle gear | (∗8) |
| +25 | Coolant (1=ON, 2=OFF) | |
| +26 | Unused | |
| +27 | Thread angle | |
| +28 | Screw lead | |
| +29 | Cut type (1) | (∗19) |

| | | |
|---|---|---|
| +30 | Cut type (2) <Ver. 1> | (∗20) |
| +31 | Number of threads | |
| +32 | Spark out | |
| +33 | Height of threads | |
| +34 | ●Chamfering (1=ON, 2=OFF) | |
| +35 | ●Cutting number/Cutting depth | (∗24) |
| +36 | ●Cutting number | |
| +37 | ●Thread type | (∗25) |
| +38 | ●Thread number | |
| +39 | Unused | |
| +40 | Unused | |
| +41 | Unused | |
| +42 | Unused | |
| +43 | Unused | |
| +44 | ●Pass point 1 X | |
| +45 | ●Pass point 1 Z | |
| +46 | ●Pass point 2 X | |
| +47 | ●Pass point 2 Z | |
| +48 | Run hour | |
| +49 | Unused | |
| +50 | Surface speed/speed selection | (∗9) |
| +51 | Speed | |
| +52 | * First variable number for roughing | |
| +53 | * First variable number for finishing | |
| +54 | * First variable number for chamfering | |
| +55 | * First variable number of a figure block | |
| +56 | * Unused | |
| +57 | * Use status flag (0: Not used, 1: Used) | |
| +58 | * First variable number of the preceding process | |
| +59 | * First variable number of the next process | |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

**NOTE**
Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

● Grooving (rough machining)

| | |
|---|---|
| + 0 | Process number |
| + 1 | Unused |
| + 2 | System utilization area (*3) |
| + 3 | Machining type (*4) |
| + 4 | Machining area (*5) |
| + 5 | Tool post (Spindle axis selection) (*6) |
| + 6 | Execution cycle (*7) |
| + 7 | Machining start point X |
| + 8 | Machining start point Z |
| + 9 | ●Program override |
| +10 | Unused |
| +11 | Surface roughness <trapezoid> |
| +12 | Unused |
| +13 | Unused |
| +14 | Tool number |
| +15 | T code |
| +16 | Unused |
| +17 | Unused |
| +18 | Cutting speed |
| +19 | Feed amount |
| +20 | Direction of rotation (1=CW, 2=CCW) |
| +21 | Cut depth |
| +22 | Finishing allowance X <trapezoid> |
| +23 | Finishing allowance Z <trapezoid> |
| +24 | Spindle gear (*8) |
| +25 | Coolant (1=ON, 2=OFF) |
| +26 | Unused |
| +27 | Grooving tool program point (*18) |
| +28 | Grooving pattern (*10) |
| +29 | Minimum groove width |

| | |
|---|---|
| +30 | Groove angle <slanted> |
| +31 | Unused |
| +32 | Start point X <ordinary, slanted, thread> |
| +33 | Start point Z <ordinary, slanted, thread> |
| +34 | Groove width <ordinary, slanted, thread> |
| +35 | Groove diameter/depth <ordinary, slanted, thread> |
| +36 | Pitch |
| +37 | Number of grooves |
| +38 | Chamfer <ordinary, thread> |
| +39 | End point X or Z <ordinary, thread> |
| +40 | Selection of groove diameter or groove depth <ordinary, slanted> (*17) |
| +41 | ●Dwell time |
| +42 | Unused |
| +43 | Unused |
| +44 | ●Pass point 1 X |
| +45 | ●Pass point 1 Z |
| +46 | ●Pass point 2 X |
| +47 | ●Pass point 2 Z |
| +48 | Run hour |
| +49 | Unused |
| +50 | Surface speed/speed selection (*9) |
| +51 | Speed |
| +52 | * First variable number for roughing |
| +53 | * First variable number for finishing |
| +54 | * First variable number for chamfering |
| +55 | * First variable number of a figure block |
| +56 | * First variable number of data block used to input a trapezoidal groove |
| +57 | * Use status flag (0: Not used, 1: Used) |
| +58 | * First variable number of the preceding process |
| +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (*?): See the note below.

**NOTE**

Thread groove can be used Ver.2 or later. Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

- Grooving (finishing-trapezoid only)

| | | | |
|---|---|---|---|
| + 0 | Process number | +30 | Unused |
| + 1 | Unused | +31 | Unused |
| + 2 | System utilization area (∗3) | +32 | Unused |
| + 3 | Machining type (∗4) | +33 | Unused |
| + 4 | Machining area (∗5) | +34 | Unused |
| + 5 | Tool post (Spindle axis selection) (∗6) | +35 | Unused |
| + 6 | Execution cycle (∗7) | +36 | Pitch |
| + 7 | Machining start point X | +37 | Number of grooves |
| + 8 | Machining start point Z | +38 | Unused |
| + 9 | Unused | +39 | Unused |
| +10 | Unused | +40 | Unused |
| +11 | Surface roughness | +41 | Unused |
| +12 | Unused | +42 | Unused |
| +13 | Unused | +43 | Unused |
| +14 | Tool number | +44 | ●Pass point 1 X |
| +15 | T code | +45 | ●Pass point 1 Z |
| +16 | Unused | +46 | ●Pass point 2 X |
| +17 | Unused | +47 | ●Pass point 2 Z |
| +18 | Cutting speed | +48 | Run hour |
| +19 | Feed amount | +49 | Unused |
| +20 | Direction of rotation (1=CW, 2=CCW) | +50 | Surface speed/speed selection (∗9) |
| +21 | Unused | +51 | Speed |
| +22 | Finishing allowance X | +52 | * First variable number for roughing |
| +23 | Finishing allowance Z | +53 | * First variable number for finishing |
| +24 | Spindle gear (∗8) | +54 | * First variable number for chamfering |
| +25 | Coolant (1=ON, 2=OFF) | +55 | * First variable number of a figure block |
| +26 | Unused | +56 | * First variable number of data block used to input a trapezoidal groove |
| +27 | Grooving tool program point (∗18) | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Grooving pattern (∗10) | +58 | * First variable number of the preceding process |
| +29 | Minimum groove width | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

**NOTE**
Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

- Grooving (input data-trapezoid and thread groove)

| | | | |
|---|---|---|---|
| + 0 | Start point X | +30 | Unused |
| + 1 | Start point Z | +31 | Unused |
| + 2 | Point 1 X | +32 | Unused |
| + 3 | Point 1 Z | +33 | Unused |
| + 4 | Round | +34 | Unused |
| + 5 | Chamfer | +35 | Unused |
| + 6 | Point 2 X | +36 | Unused |
| + 7 | Point 2 Z | +37 | Unused |
| + 8 | Round | +38 | Unused |
| + 9 | Chamfer | +39 | Unused |
| +10 | Point 3 X | +40 | Unused |
| +11 | Point 3 Z | +41 | Unused |
| +12 | Round | +42 | Unused |
| +13 | Chamfer | +43 | Unused |
| +14 | Point 4 X | +44 | ●Pass point 1 X |
| +15 | Point 4 Z | +45 | ●Pass point 1 Z |
| +16 | Round | +46 | ●Pass point 2 X |
| +17 | Chamfer | +47 | ●Pass point 2 Z |
| +18 | End point X | +48 | Run hour |
| +19 | End point Z | +49 | Unused |
| +20 | Unused | +50 | Surface speed/speed selection (∗9) |
| +21 | Unused | +51 | Speed |
| +22 | Unused | +52 | * First variable number for roughing |
| +23 | Unused | +53 | * First variable number for finishing |
| +24 | Unused | +54 | * First variable number for chamfering |
| +25 | Unused | +55 | * First variable number of a figure block |
| +26 | Unused | +56 | * First variable number of data block used to input a trapezoidal groove |
| +27 | Unused | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Unused | +58 | * First variable number of the preceding process |
| +29 | Unused | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

**NOTE**
Thread groove can be used Ver.2 or later. Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

- Necking

| | | | |
|---|---|---|---|
| + 0 | Process number | +30 | Necking dimension D (radius) |
| + 1 | Unused | +31 | Tool angle of the tool used (∗27) |
| + 2 | System utilization area (∗3) | +32 | Cutting edge angle of the tool used (∗27) |
| + 3 | Machining type (∗4) | +33 | Necking figure (∗11) |
| + 4 | Machining area (∗5) | +34 | Standard diameter (diameter) |
| + 5 | Tool post (Spindle axis selection) (∗6) | +35 | Width (radius) |
| + 6 | Machining cycle (∗7) | +36 | Depth (radius) |
| + 7 | Machining start point X | +37 | Corner radius |
| + 8 | Machining start point Z | +38 | Approach angle |
| + 9 | ●Program override | +39 | Relief amount (radius) |
| +10 | Unused | +40 | Relief angle (radius) |
| +11 | Surface roughness | +41 | Unused |
| +12 | Unused | +42 | Unused |
| +13 | Unused | +43 | Unused |
| +14 | Tool number | +44 | ●Pass point 1 X |
| +15 | T code | +45 | ●Pass point 1 Z |
| +16 | Unused | +46 | ●Pass point 2 X |
| +17 | Unused | +47 | ●Pass point 2 Z |
| +18 | Cutting speed | +48 | Run hour |
| +19 | Feed amount | +49 | Unused |
| +20 | Direction of rotation (1=CW, 2=CCW) | +50 | Surface speed/speed selection (∗9) |
| +21 | Unused | +51 | Speed |
| +22 | Unused | +52 | * First variable number for roughing |
| +23 | Unused | +53 | * First variable number for finishing |
| +24 | Spindle gear (∗8) | +54 | * First variable number for chamfering |
| +25 | Coolant (1=ON, 2=OFF) | +55 | * First variable number of a figure block |
| +26 | Unused | +56 | * Unused |
| +27 | Necking dimention A (radius) | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Necking dimension B (radius) | +58 | * First variable number of the preceding process |
| +29 | Necking dimension C (radius) | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

**NOTE**
Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

- Center Drilling

| | | | |
|---|---|---|---|
| + 0 | Process number | +30 | Unused |
| + 1 | Unused | +31 | Unused |
| + 2 | System utilization area (∗3) | +32 | Unused |
| + 3 | Machining type (∗4) | +33 | Unused |
| + 4 | Unused | +34 | Unused |
| + 5 | Tool post (Spindle axis selection) (∗6) | +35 | Unused |
| + 6 | Machining cycle (∗7) | +36 | ●Program override |
| + 7 | Cutting start point X | +37 | Unused |
| + 8 | Cutting start point Z | +38 | Unused |
| + 9 | ■Start point Z | +39 | Unused |
| +10 | Unused | +40 | Unused |
| +11 | Unused | +41 | Unused |
| +12 | Unused | +42 | Unused |
| +13 | Unused | +43 | Automatic preceding process determination flag (∗12) |
| +14 | Tool number | +44 | ●Pass point 1 X |
| +15 | T code | +45 | ●Pass point 1 Z |
| +16 | Unused | +46 | ●Pass point 2 X |
| +17 | Unused | +47 | ●Pass point 2 Z |
| +18 | Cutting speed | +48 | Run hour |
| +19 | Feed amount | +49 | Unused |
| +20 | Direction of rotation (1=CW, 2=CCW) | +50 | Surface speed/speed selection (∗9) |
| +21 | Unused | +51 | Speed |
| +22 | Unused | +52 | * First variable number for roughing |
| +23 | Unused | +53 | * First variable number for finishing |
| +24 | Spindle gear (∗8) | +54 | * First variable number for chamfering |
| +25 | Coolant (1=ON, 2=OFF) | +55 | * First variable number of a figure block |
| +26 | ■End point Z/Hole depth (∗28) | +56 | * Unused |
| +27 | Hole bottom point | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Hole diameter/Chamfer diameter (Nominal diameter) | +58 | * First variable number of the preceding process |
| +29 | Dwell time at hole bottom point | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

**NOTE**
Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

● Drilling

| | | | |
|---|---|---|---|
| + 0 | Process number | +30 | Machining pattern (∗14) |
| + 1 | Machining type (2) (∗13) | +31 | Decrement in depth of cut |
| + 2 | System utilization area (∗3) | +32 | Relief return amount |
| + 3 | Machining type (∗4) | +33 | Minimum value for the depth of cut |
| + 4 | Unused | +34 | Residual point Z/Chamfer length/Shift amount |
| + 5 | Tool post (Spindle axis selection) (∗6) | +35 | ■FEED/Rev 2/Return speed |
| + 6 | Machining cycle (∗7) | +36 | ■Start feedrate |
| + 7 | Cutting start point X | +37 | ■Start clearance |
| + 8 | Cutting start point Z | +38 | ■End feedrate |
| + 9 | ■Start point Z | +39 | ■End clearance |
| +10 | Override amount | +40 | Unused |
| +11 | Unused | +41 | Unused |
| +12 | Unused | +42 | Unused |
| +13 | Unused | +43 | Automatic preceding process determination flag (∗12) |
| +14 | Tool number | +44 | ●Pass point 1 X |
| +15 | T code | +45 | ●Pass point 1 Z |
| +16 | Unused | +46 | ●Pass point 2 X |
| +17 | Unused | +47 | ●Pass point 2 Z |
| +18 | Cutting speed | +48 | Run hour |
| +19 | Feed amount | +49 | Unused |
| +20 | Direction of rotation (1=CW, 2=CCW) | +50 | Surface speed/speed selection (∗9) |
| +21 | Cut depth | +51 | Speed |
| +22 | Unused | +52 | * First variable number for roughing |
| +23 | Unused | +53 | * First variable number for finishing |
| +24 | Spindle gear (∗8) | +54 | * First variable number for chamfering |
| +25 | Coolant (1=ON, 2=OFF) | +55 | * First variable number of a figure block |
| +26 | ■End point Z/Hole depth (∗28) | +56 | * Unused |
| +27 | Hole bottom point | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Hole diameter | +58 | * First variable number of the preceding process |
| +29 | Dwell time at hole bottom point | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

**NOTE**
Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

- Tapping

| | | | |
|---|---|---|---|
| + 0 | Process number | +30 | Pitch |
| + 1 | Unused | +31 | Unused |
| + 2 | System utilization area (∗3) | +32 | Unused |
| + 3 | Machining type (∗4) | +33 | Unused |
| + 4 | Unused | +34 | Unused |
| + 5 | Tool post (Spindle axis selection) (∗6) | +35 | Unused |
| + 6 | Machining cycle (∗7) | +36 | Unused |
| + 7 | Cutting start point X | +37 | Unused |
| + 8 | Cutting start point Z | +38 | Unused |
| + 9 | ■Start point Z | +39 | Unused |
| +10 | Unused | +40 | Unused |
| +11 | Unused | +41 | Unused |
| +12 | Unused | +42 | Unused |
| +13 | Unused | +43 | Automatic preceding process determination flag (∗12) |
| +14 | Tool number | +44 | ●Pass point 1 X |
| +15 | T code | +45 | ●Pass point 1 Z |
| +16 | Unused | +46 | ●Pass point 2 X |
| +17 | Unused | +47 | ●Pass point 2 Z |
| +18 | Cutting speed | +48 | Run hour |
| +19 | Feed amount | +49 | Unused |
| +20 | Direction of rotation (1=CW, 2=CCW) | +50 | Surface speed/speed selection (∗9) |
| +21 | Unused | +51 | Speed |
| +22 | Unused | +52 | * First variable number for roughing |
| +23 | Unused | +53 | * First variable number for finishing |
| +24 | Spindle gear (∗8) | +54 | * First variable number for chamfering |
| +25 | Coolant (1=ON, 2=OFF) | +55 | * First variable number of a figure block |
| +26 | ■End point Z/Hole depth (∗28) | +56 | * Unused |
| +27 | Hole bottom point | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Nominal diameter | +58 | * First variable number of the preceding process |
| +29 | Dwell time at hole bottom point | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

**NOTE**
Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

● Single Action (when bit 0 of parameter No.9766 is 0)

| | | | |
|---|---|---|---|
| + 0 | Process number | +30 | Unused |
| + 1 | Unused | +31 | Unused |
| + 2 | System utilization area (∗3) | +32 | Unused |
| + 3 | Machining type (∗4) | +33 | Unused |
| + 4 | Unused | +34 | Unused |
| + 5 | Tool post (Spindle axis selection) (∗6) | +35 | Unused |
| + 6 | Execution cycle (∗7) | +36 | Unused |
| + 7 | Cutting start point X | +37 | Unused |
| + 8 | Cutting start point Z | +38 | Unused |
| + 9 | Unused | +39 | Unused |
| +10 | Unused | +40 | Unused |
| +11 | Unused | +41 | Unused |
| +12 | Unused | +42 | Unused |
| +13 | Unused | +43 | Unused |
| +14 | Tool number | +44 | ●Pass point 1 X |
| +15 | T code | +45 | ●Pass point 1 Z |
| +16 | Unused | +46 | ●Pass point 2 X |
| +17 | Unused | +47 | ●Pass point 2 Z |
| +18 | Cutting speed | +48 | Run hour |
| +19 | Unused | +49 | Unused |
| +20 | Unused | +50 | Surface speed/speed selection (∗9) |
| +21 | Unused | +51 | Speed |
| +22 | Unused | +52 | * First variable number for roughing |
| +23 | Unused | +53 | * First variable number for finishing |
| +24 | Unused | +54 | * First variable number for chamfering |
| +25 | Unused | +55 | * First variable number of a figure block |
| +26 | Unused | +56 | * Unused |
| +27 | Feedrate (1=mm/rev., 2=mm/min.) | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Unused | +58 | * First variable number of the preceding process |
| +29 | Unused | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

**NOTE**
Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

● Single Action II (when bit 0 of parameter No.9766 is 0) <available on Ver.3>

| | | | |
|---|---|---|---|
| + 0 | Process number | +30 | Unused |
| + 1 | Unused | +31 | Unused |
| + 2 | System utilization area (∗3) | +32 | Unused |
| + 3 | Machining type (∗4) | +33 | Unused |
| + 4 | Type (∗29) | +34 | Unused |
| + 5 | Tool post (Spindle axis selection) (∗6) | +35 | Unused |
| + 6 | Execution cycle (∗7) | +36 | Unused |
| + 7 | Unused | +37 | Unused |
| + 8 | Unused | +38 | Unused |
| + 9 | Unused | +39 | Unused |
| +10 | Unused | +40 | Unused |
| +11 | Unused | +41 | Unused |
| +12 | Unused | +42 | Unused |
| +13 | Unused | +43 | Unused |
| +14 | Tool number | +44 | Unused |
| +15 | T code | +45 | Unused |
| +16 | Unused | +46 | Unused |
| +17 | Unused | +47 | Unused |
| +18 | Unused | +48 | Run hour |
| +19 | Unused | +49 | Unused |
| +20 | Unused | +50 | Unused |
| +21 | Unused | +51 | Unused |
| +22 | Unused | +52 | * First variable number for roughing |
| +23 | Unused | +53 | * First variable number for finishing |
| +24 | Unused | +54 | * First variable number for chamfering |
| +25 | Unused | +55 | * First variable number of a figure block |
| +26 | Unused | +56 | * Unused |
| +27 | Unused | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Unused | +58 | * First variable number of the preceding process |
| +29 | Unused | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

**NOTE**
Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

- Calling Subprograms (when bit 3 of parameter No.9771 is 0)

| | | | |
|---|---|---|---|
| + 0 | Process number | +30 | Data 3 |
| + 1 | Unused | +31 | Data 4 |
| + 2 | System utilization area (∗3) | +32 | Unused |
| + 3 | Machining type (∗4) | +33 | Unused |
| + 4 | Unused | +34 | Unused |
| + 5 | Tool post (∗6) | +35 | Unused |
| + 6 | Execution cycle (∗7) | +36 | Unused |
| + 7 | Unused | +37 | Unused |
| + 8 | Unused | +38 | Unused |
| + 9 | Unused | +39 | Unused |
| +10 | Unused | +40 | Unused |
| +11 | Unused | +41 | Unused |
| +12 | Unused | +42 | Unused |
| +13 | Unused | +43 | Unused |
| +14 | Unused | +44 | Unused |
| +15 | Unused | +45 | Unused |
| +16 | Unused | +46 | Unused |
| +17 | Unused | +47 | Unused |
| +18 | Unused | +48 | Run hour |
| +19 | Unused | +49 | Unused |
| +20 | Unused | +50 | Unused |
| +21 | Unused | +51 | Unused |
| +22 | Unused | +52 | * First variable number for roughing |
| +23 | Unused | +53 | * First variable number for finishing |
| +24 | Unused | +54 | * First variable number for chamfering |
| +25 | Unused | +55 | * First variable number of a figure block |
| +26 | Unused | +56 | * Unused |
| +27 | Sub program | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Data 1 | +58 | * First variable number of the preceding process |
| +29 | Data 2 | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

**NOTE**
Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

● Calling Subprograms II (when bit 3 of parameter No.9771 is 1) <available on Ver.2 or later>

| | | | |
|---|---|---|---|
| + 0 | Process number | +30 | Unused |
| + 1 | Unused | +31 | Unused |
| + 2 | System utilization area (∗3) | +32 | Data Z |
| + 3 | Machining type (∗4) | +33 | Sub program No. |
| + 4 | Unused | +34 | Unused |
| + 5 | Tool post (∗6) | +35 | Unused |
| + 6 | Execution cycle (∗7) | +36 | Unused |
| + 7 | Data A | +37 | Unused |
| + 8 | Data B | +38 | Unused |
| + 9 | Data C | +39 | Unused |
| +10 | Data I | +40 | Unused |
| +11 | Data J | +41 | Unused |
| +12 | Data K | +42 | Unused |
| +13 | Data D | +43 | Unused |
| +14 | Data E | +44 | Unused |
| +15 | Data F | +45 | Unused |
| +16 | Data H | +46 | Unused |
| +17 | Data M | +47 | Unused |
| +18 | Data Q | +48 | Run hour |
| +19 | Data R | +49 | Unused |
| +20 | Data S | +50 | Unused |
| +21 | Data T | +51 | Unused |
| +22 | Data U | +52 | * First variable number for roughing |
| +23 | Data V | +53 | * First variable number for finishing |
| +24 | Data W | +54 | * First variable number for chamfering |
| +25 | Data X | +55 | * First variable number of a figure block |
| +26 | Data Y | +56 | * Unused |
| +27 | Unused | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Unused | +58 | * First variable number of the preceding process |
| +29 | Unused | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

**NOTE**
Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

- C-axis Center Drilling

| | | | |
|---|---|---|---|
| + 0 | Process number | +30 | Unused |
| + 1 | Unused | +31 | Unused |
| + 2 | System utilization area (∗3) | +32 | Unused |
| + 3 | Machining type (∗4) | +33 | Unused |
| + 4 | Machining area (∗5) | +34 | Unused |
| + 5 | Tool post (Spindle axis selection) (∗6) | +35 | Unused |
| + 6 | Machining cycle (∗7) | +36 | ●Program override |
| + 7 | Machining start point X | +37 | Unused |
| + 8 | Machining start point Z | +38 | Unused |
| + 9 | Unused | +39 | Unused |
| +10 | Unused | +40 | Unused |
| +11 | Unused | +41 | Unused |
| +12 | Unused | +42 | Unused |
| +13 | Unused | +43 | Automatic preceding process determination flag (∗12) |
| +14 | Tool number | +44 | ■Pass point 1 X |
| +15 | T code | +45 | ■Pass point 1 Z |
| +16 | Unused | +46 | ■Pass point 2 X |
| +17 | Unused | +47 | ■Pass point 2 Z |
| +18 | Tool speed (rpm) | +48 | Run hour |
| +19 | Feedrate | +49 | Unused |
| +20 | Direction of rotation (1=CW, 2=CCW) | +50 | Unused |
| +21 | Unused | +51 | Unused |
| +22 | Unused | +52 | * First variable number for roughing |
| +23 | Unused | +53 | * First variable number for finishing |
| +24 | Milling gear | +54 | * First variable number for chamfering |
| +25 | Coolant (1=ON, 2=OFF) | +55 | * First variable number of a figure block |
| +26 | Unused | +56 | * Unused |
| +27 | Hole pattern (1=equal, 2=unequal) | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Hole diameter/chamfer diameterer | +58 | * First variable number of the preceding process |
| +29 | Dwell time at hole bottom point | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

**NOTE**
Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

- C-axis Drilling

| | | | |
|---|---|---|---|
| + 0 | Process number | +30 | Machining pattern (∗14) |
| + 1 | Machining type (2) (∗13) | +31 | Decrement in depth of cut |
| + 2 | System utilization area (∗3) | +32 | Relief return amount |
| + 3 | Machining type (∗4) | +33 | Minimum value for the depth of cut |
| + 4 | Machining area (∗5) | +34 | ■Chamfer length/shift amount |
| + 5 | Tool post (Spindle axis selection) (∗6) | +35 | ■FEED/Rev 2/return speed/shift direction (∗26) |
| + 6 | Machining cycle (∗7) | +36 | ■Start feedrate |
| + 7 | Machining start point X | +37 | ■Start clearance |
| + 8 | Machining start point Z | +38 | ■End feedrate |
| + 9 | ●Program override | +39 | ■End clearance |
| +10 | Unused | +40 | Unused |
| +11 | Unused | +41 | Unused |
| +12 | Unused | +42 | Unused |
| +13 | Unused | +43 | Automatic preceding process determination flag (∗12) |
| +14 | Tool number | +44 | ●Pass point 1 X |
| +15 | T code | +45 | ●Pass point 1 Z |
| +16 | Unused | +46 | ●Pass point 2 X |
| +17 | Unused | +47 | ●Pass point 2 Z |
| +18 | Tool speed | +48 | Run hour |
| +19 | Feedrate | +49 | Unused |
| +20 | ■Direction of rotation (1=CW, 2=CCW) | +50 | Surface speed/speed selection (∗9) |
| +21 | Cutting depth | +51 | Speed |
| +22 | ■Orientation M | +52 | * First variable number for roughing |
| +23 | Unused | +53 | * First variable number for finishing |
| +24 | Milling gear | +54 | * First variable number for chamfering |
| +25 | Coolant (1=ON, 2=OFF) | +55 | * First variable number of a figure block |
| +26 | Unused | +56 | * Unused |
| +27 | Hole pattern (1=equal, 2=unequal) | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Hole diameter | +58 | * First variable number of the preceding process |
| +29 | Dwell time at hole bottom point | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

**NOTE**
Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

- C-axis Tapping

| | | | |
|---|---|---|---|
| + 0 | Process number | +30 | Pitch |
| + 1 | Unused | +31 | Unused |
| + 2 | System utilization area (∗3) | +32 | Unused |
| + 3 | Machining type (∗4) | +33 | Unused |
| + 4 | Machining area (∗5) | +34 | Unused |
| + 5 | Tool post (Spindle axis selection) (∗6) | +35 | Unused |
| + 6 | Machining cycle (∗7) | +36 | Unused |
| + 7 | Machining start point X | +37 | Unused |
| + 8 | Machining start point Z | +38 | Unused |
| + 9 | Unused | +39 | Unused |
| +10 | Unused | +40 | Unused |
| +11 | Unused | +41 | Unused |
| +12 | Unused | +42 | Unused |
| +13 | Unused | +43 | Automatic preceding process determination flag (∗12) |
| +14 | Tool number | +44 | ●Pass point 1 X |
| +15 | T code | +45 | ●Pass point 1 Z |
| +16 | Unused | +46 | ●Pass point 2 X |
| +17 | Unused | +47 | ●Pass point 2 Z |
| +18 | Tool speed (rpm) | +48 | Run hour |
| +19 | Feedrate | +49 | Unused |
| +20 | ■Direction of rotation (1=CW, 2=CCW) | +50 | Unused |
| +21 | Unused | +51 | Unused |
| +22 | Unused | +52 | * First variable number for roughing |
| +23 | Unused | +53 | * First variable number for finishing |
| +24 | Milling gear | +54 | * First variable number for chamfering |
| +25 | Coolant (1=ON, 2=OFF) | +55 | * First variable number of a figure block |
| +26 | Unused | +56 | * Unused |
| +27 | Hole pattern (1=equal, 2=unequal) | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Nominal diameter | +58 | * First variable number of the preceding process |
| +29 | Dwell time at hole bottom point | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

**NOTE**
Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

● C-axis Grooving (rough machining)

| | | | |
|---|---|---|---|
| + 0 | Process number | +30 | Unused |
| + 1 | Unused | +31 | Unused |
| + 2 | System utilization area (∗3) | +32 | Unused |
| + 3 | Machining type (∗4) | +33 | Unused |
| + 4 | Machining area (∗5) | +34 | Unused |
| + 5 | Tool post (Spindle axis selection) (∗6) | +35 | Unused |
| + 6 | Machining cycle (∗7) | +36 | ●Program override |
| + 7 | Machining start point X | +37 | Unused |
| + 8 | Machining start point Z | +38 | Unused |
| + 9 | Unused | +39 | Unused |
| +10 | Unused | +40 | Unused |
| +11 | Unused | +41 | Unused |
| +12 | Unused | +42 | Unused |
| +13 | Chamfer amount (∗15) | +43 | Unused |
| +14 | Tool number | +44 | ●Pass point 1 X |
| +15 | T code | +45 | ●Pass point 1 Z |
| +16 | Unused | +46 | ●Pass point 2 X |
| +17 | Unused | +47 | ●Pass point 2 Z |
| +18 | Tool speed (rpm) | +48 | Run hour |
| +19 | Feedrate-1 | +49 | Unused |
| +20 | Milling gear | +50 | Unused |
| +21 | Feedrate-2 | +51 | Unused |
| +22 | Unused | +52 | * First variable number for roughing |
| +23 | Unused | +53 | * First variable number for finishing |
| +24 | Unused | +54 | * First variable number for chamfering |
| +25 | Coolant (1=ON, 2=OFF) | +55 | * First variable number of a figure block |
| +26 | Unused | +56 | * Unused |
| +27 | Groove shape (1=regular, 2=irregular) | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Groove diameter | +58 | * First variable number of the preceding process |
| +29 | Unused | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

**NOTE**
Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

- C-axis Grooving (chamfering)

| | | | |
|---|---|---|---|
| + 0 | Process number | +30 | Unused |
| + 1 | Unused | +31 | Unused |
| + 2 | System utilization area (∗3) | +32 | Unused |
| + 3 | Machining type (∗4) | +33 | Unused |
| + 4 | Machining area (∗5) | +34 | Unused |
| + 5 | Tool post (Spindle axis selection) (∗6) | +35 | Unused |
| + 6 | Machining cycle (∗7) | +36 | ●Program override |
| + 7 | Machining start point X | +37 | Unused |
| + 8 | Machining start point Z | +38 | Unused |
| + 9 | Unused | +39 | Unused |
| +10 | Unused | +40 | Unused |
| +11 | Unused | +41 | Unused |
| +12 | Unused | +42 | Unused |
| +13 | Chamfer (∗15) | +43 | Unused |
| +14 | Tool number | +44 | ●Pass point 1 X |
| +15 | T code | +45 | ●Pass point 1 Z |
| +16 | Unused | +46 | ●Pass point 2 X |
| +17 | Unused | +47 | ●Pass point 2 Z |
| +18 | Tool speed (rpm) | +48 | Run hour |
| +19 | Feedrate | +49 | Unused |
| +20 | Milling gear | +50 | Unused |
| +21 | Unused | +51 | Unused |
| +22 | Unused | +52 | * First variable number for roughing |
| +23 | Unused | +53 | * First variable number for finishing |
| +24 | Unused | +54 | * First variable number for chamfering |
| +25 | Coolant (1=ON, 2=OFF) | +55 | * First variable number of a figure block |
| +26 | Unused | +56 | * Unused |
| +27 | Groove shape (1=regular, 2=irregular) | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Groove diameter | +58 | * First variable number of the preceding process |
| +29 | Unused | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

**NOTE**
Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

● C-axis Notching (rough machining)

| | | | |
|---|---|---|---|
| + 0 | Process number | +30 | Unused |
| + 1 | Unused | +31 | Unused |
| + 2 | System utilization area (∗3) | +32 | Unused |
| + 3 | Machining type (∗4) | +33 | Unused |
| + 4 | Machining area (∗5) | +34 | Unused |
| + 5 | Tool post (Spindle axis selection) (∗6) | +35 | Unused |
| + 6 | Machining cycle (∗7) | +36 | ●Program override |
| + 7 | Unused | +37 | Unused |
| + 8 | Start point Z | +38 | Unused |
| + 9 | End point Z | +39 | Unused |
| +10 | Unused | +40 | Unused |
| +11 | Unused | +41 | Unused |
| +12 | Cutting allowance X | +42 | Unused |
| +13 | Chamfer (∗15) | +43 | Unused |
| +14 | Tool number | +44 | ●Pass point 1 X |
| +15 | T code | +45 | ●Pass point 1 Z |
| +16 | Unused | +46 | ●Pass point 2 X |
| +17 | Unused | +47 | ●Pass point 2 Z |
| +18 | Tool speed | +48 | Run hour |
| +19 | Feedrate | +49 | System utilization area (∗3) |
| +20 | Milling gear | +50 | Unused |
| +21 | Depth of cut | +51 | Unused |
| +22 | Finishing allowance X | +52 | * First variable number for roughing |
| +23 | Finishing allowance Z | +53 | * First variable number for finishing |
| +24 | Unused | +54 | * First variable number for chamfering |
| +25 | Coolant (1=ON, 2=OFF) | +55 | * First variable number of a figure block |
| +26 | Unused | +56 | * Unused |
| +27 | Unused | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Unused | +58 | * First variable number of the preceding process |
| +29 | Unused | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

**NOTE**

Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

- C-axis Notching (finish machining)

| | | | |
|---|---|---|---|
| + 0 | Process number | +30 | Unused |
| + 1 | Unused | +31 | Unused |
| + 2 | System utilization area (∗3) | +32 | Unused |
| + 3 | Machining type (∗4) | +33 | Unused |
| + 4 | Machining area (∗5) | +34 | Unused |
| + 5 | Tool post (Spindle axis selection) (∗6) | +35 | Unused |
| + 6 | Machining cycle (∗7) | +36 | ●Program override |
| + 7 | Unused | +37 | Unused |
| + 8 | Start point Z | +38 | Unused |
| + 9 | End point Z | +39 | Unused |
| +10 | Unused | +40 | Unused |
| +11 | Unused | +41 | Unused |
| +12 | Cutting allowance X | +42 | Unused |
| +13 | Chamfer (∗15) | +43 | Unused |
| +14 | Tool number | +44 | ●Pass point 1 X |
| +15 | T code | +45 | ●Pass point 1 Z |
| +16 | Unused | +46 | ●Pass point 2 X |
| +17 | Unused | +47 | ●Pass point 2 Z |
| +18 | Tool speed | +48 | Run hour |
| +19 | Feedrate | +49 | System utilization area (∗3) |
| +20 | Milling gear | +50 | Unused |
| +21 | Unused | +51 | Unused |
| +22 | Finishing allowance X | +52 | * First variable number for roughing |
| +23 | Finishing allowance Z | +53 | * First variable number for finishing |
| +24 | Unused | +54 | * First variable number for chamfering |
| +25 | Coolant (1=ON, 2=OFF) | +55 | * First variable number of a figure block |
| +26 | Unused | +56 | * Unused |
| +27 | Unused | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Unused | +58 | * First variable number of the preceding process |
| +29 | Unused | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

**NOTE**
Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

- C-axis Notching (chamfering)

| | | | |
|---|---|---|---|
| + 0 | Process number | +30 | Unused |
| + 1 | Unused | +31 | Unused |
| + 2 | System utilization area (∗3) | +32 | Unused |
| + 3 | Machining type (∗4) | +33 | Unused |
| + 4 | Machining area (∗5) | +34 | Unused |
| + 5 | Tool post (Spindle axis selection) (∗6) | +35 | Unused |
| + 6 | Machining cycle (∗7) | +36 | ●Program override |
| + 7 | Unused | +37 | Unused |
| + 8 | Start point Z | +38 | Unused |
| + 9 | End point Z | +39 | Unused |
| +10 | Unused | +40 | Unused |
| +11 | Unused | +41 | Unused |
| +12 | Unused | +42 | Unused |
| +13 | Chamfer amount (∗15) | +43 | Unused |
| +14 | Tool number | +44 | ●Pass point 1 X |
| +15 | T code | +45 | ●Pass point 1 Z |
| +16 | Unused | +46 | ●Pass point 2 X |
| +17 | Unused | +47 | ●Pass point 2 Z |
| +18 | Tool speed (rpm) | +48 | Run hour |
| +19 | Feedrate | +49 | System utilization area (∗3) |
| +20 | Milling gear | +50 | Unused |
| +21 | Unused | +51 | Unused |
| +22 | Unused | +52 | * First variable number for roughing |
| +23 | Unused | +53 | * First variable number for finishing |
| +24 | Unused | +54 | * First variable number for chamfering |
| +25 | Coolant (1=ON, 2=OFF) | +55 | * First variable number of a figure block |
| +26 | Unused | +56 | * Unused |
| +27 | Unused | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Unused | +58 | * First variable number of the preceding process |
| +29 | Unused | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

**NOTE**
Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

- C-axis Cylindrical Machining (rough machining)

| | | | |
|---|---|---|---|
| + 0 | Process number | +30 | Development drawing : Z-axis end coordinate (∗16) |
| + 1 | Unused | +31 | Development drawing : C-axis diameter (∗16) |
| + 2 | System utilization area (∗3) | +32 | Unused |
| + 3 | Machining type (∗4) | +33 | Unused |
| + 4 | Unused | +34 | Unused |
| + 5 | Tool post (Spindle axis selection) (∗6) | +35 | Unused |
| + 6 | Machining cycle (∗7) | +36 | ●Program override |
| + 7 | Grooving start point X | +37 | Unused |
| + 8 | Unused | +38 | Unused |
| + 9 | Unused | +39 | Unused |
| +10 | Unused | +40 | Unused |
| +11 | Unused | +41 | Unused |
| +12 | Unused | +42 | Unused |
| +13 | Chamfer amount (∗15) | +43 | Unused |
| +14 | Tool number | +44 | ●Pass point 1 X |
| +15 | T code | +45 | ●Pass point 1 Z |
| +16 | Unused | +46 | ●Pass point 2 X |
| +17 | Unused | +47 | ●Pass point 2 Z |
| +18 | Tool speed (rpm) | +48 | Run hour |
| +19 | Feedrate-1 | +49 | System utilization area (∗3) |
| +20 | Milling gear | +50 | Unused |
| +21 | Feedrate-2 | +51 | Unused |
| +22 | Unused | +52 | * First variable number for roughing |
| +23 | Unused | +53 | * First variable number for finishing |
| +24 | Unused | +54 | * First variable number for chamfering |
| +25 | Coolant (1=ON, 2=OFF) | +55 | * First variable number of a figure block |
| +26 | Unused | +56 | * Unused |
| +27 | Depth of the groove | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Groove diameter | +58 | * First variable number of the preceding process |
| +29 | Development drawing : Z-axis start coordinate (∗16) | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

**NOTE**
Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

● C-axis Cylindrical Machining (chamfering)

| | | | |
|---|---|---|---|
| + 0 | Process number | +30 | Development drawing<br>: Z-axis end coordinate (∗16) |
| + 1 | Unused | +31 | Development drawing<br>: C-axis diameter (∗16) |
| + 2 | System utilization area (∗3) | +32 | Unused |
| + 3 | Machining type (∗4) | +33 | Unused |
| + 4 | Unused | +34 | Unused |
| + 5 | Tool post (∗6) | +35 | Unused |
| + 6 | Machining cycle (∗7) | +36 | ●Program override |
| + 7 | Grooving start coordinate X | +37 | Unused |
| + 8 | Unused | +38 | Unused |
| + 9 | Unused | +39 | Unused |
| +10 | Unused | +40 | Unused |
| +11 | Unused | +41 | Unused |
| +12 | Unused | +42 | Unused |
| +13 | Chamfer amount (∗15) | +43 | Unused |
| +14 | Tool number | +44 | ●Pass point 1 X |
| +15 | T code | +45 | ●Pass point 1 Z |
| +16 | Unused | +46 | ●Pass point 2 X |
| +17 | Unused | +47 | ●Pass point 2 Z |
| +18 | Tool speed (rpm) | +48 | Run hour |
| +19 | Feedrate | +49 | System utilization area (∗3) |
| +20 | Milling gear | +50 | Unused |
| +21 | Unused | +51 | Unused |
| +22 | Unused | +52 | * First variable number for roughing |
| +23 | Unused | +53 | * First variable number for finishing |
| +24 | Unused | +54 | * First variable number for chamfering |
| +25 | Coolant (1=ON, 2=OFF) | +55 | * First variable number of a figure block |
| +26 | Unused | +56 | * Unused |
| +27 | Depth of the groove | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Groove diameter | +58 | * First variable number of the preceding process |
| +29 | Development drawing<br>: Z-axis start coordinate (∗16) | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

**NOTE**
Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

- Transfer, Auxiliary Process

| | | | |
|---|---|---|---|
| + 0 | Process number | +30 | Display data 10 (Data format) |
| + 1 | Unused | +31 | Display data 11 |
| + 2 | System utilization area (∗3) | +32 | Display data 11 (Data format) |
| + 3 | Machining type (∗4) | +33 | Display data 12 |
| + 4 | Unused | +34 | Display data 12 (Data format) |
| + 5 | Tool post (Spindle axis selection) (∗6, ∗21) | +35 | |
| + 6 | Machining cycle (∗7, ∗21) | +36 | Data area for user programs used with the macro executor (+36 to +47) |
| + 7 | Type | +37 | |
| + 8 | Type (Data format) | +38 | |
| + 9 | Head | +39 | |
| +10 | Head (Data format) | +40 | |
| +11 | Dispaly data 1 | +41 | |
| +12 | Dispaly data 1 (Data format) | +42 | |
| +13 | Display data 2 | +43 | |
| +14 | Display data 2 (Data format) | +44 | |
| +15 | Display data 3 | +45 | |
| +16 | Display data 3 (Data format) | +46 | |
| +17 | Display data 4 | +47 | |
| +18 | Display data 4 (Data format) | +48 | Run hour |
| +19 | Display data 5 | +49 | Unused |
| +20 | Display data 5 (Data format) | +50 | Surface speed/speed selection (∗9) |
| +21 | Display data 6 | +51 | Speed |
| +22 | Display data 6 (Data format) | +52 | * First variable number for roughing |
| +23 | Display data 7 | +53 | * First variable number for finishing |
| +24 | Display data 7 (Data format) | +54 | * First variable number for chamfering |
| +25 | Display data 8 | +55 | * First variable number of a figure block |
| +26 | Display data 8 (Data format) | +56 | * Unused |
| +27 | Display data 9 | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Display data 9 (Data format) | +58 | * First variable number of the preceding process |
| +29 | Display data 10 | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

**NOTE**
Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

- M-code Process

| | | | |
|---|---|---|---|
| + 0 | Process number | +30 | Unused |
| + 1 | Unused | +31 | Unused |
| + 2 | System utilization area (∗3) | +32 | Unused |
| + 3 | Machining type (∗4) | +33 | Unused |
| + 4 | Unused | +34 | Unused |
| + 5 | Tool post (Spindle axis selection) (∗6) | +35 | Unused |
| + 6 | Machining cycle (∗7) | +36 | Unused |
| + 7 | Unused | +37 | Unused |
| + 8 | Unused | +38 | Unused |
| + 9 | M-code (1) | +39 | Unused |
| +10 | M-code (2) | +40 | Unused |
| +11 | M-code (3) | +41 | Unused |
| +12 | M-code (4) | +42 | Unused |
| +13 | M-code (5) | +43 | Unused |
| +14 | Unused | +44 | Unused |
| +15 | Unused | +45 | Unused |
| +16 | Unused | +46 | Unused |
| +17 | Unused | +47 | Unused |
| +18 | Unused | +48 | Run hour |
| +19 | Unused | +49 | Unused |
| +20 | Unused | +50 | Unused |
| +21 | Unused | +51 | Unused |
| +22 | Unused | +52 | * First variable number for roughing |
| +23 | Unused | +53 | * First variable number for finishing |
| +24 | Unused | +54 | * First variable number for chamfering |
| +25 | Unused | +55 | * First variable number of a figure block |
| +26 | Unused | +56 | * Unused |
| +27 | Unused | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Unused | +58 | * First variable number of the preceding process |
| +29 | Unused | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

**NOTE**
Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

● Program End Process

| | | | |
|---|---|---|---|
| + 0 | Process number | +30 | Unused |
| + 1 | Unused | +31 | Unused |
| + 2 | System utilization area (∗3) | +32 | Unused |
| + 3 | Machining type (∗4) | +33 | Unused |
| + 4 | Unused | +34 | Unused |
| + 5 | Tool post (Spindle axis selection) (∗6) | +35 | Unused |
| + 6 | Machining cycle (∗7) | +36 | Unused |
| + 7 | Unused | +37 | Unused |
| + 8 | Unused | +38 | Unused |
| + 9 | Return code | +39 | Unused |
| +10 | Return point X | +40 | Unused |
| +11 | Return point Z | +41 | Unused |
| +12 | Return point C | +42 | Unused |
| +13 | Unused | +43 | Unused |
| +14 | End M code | +44 | Unused |
| +15 | Loop count | +45 | Unused |
| +16 | Unused | +46 | Unused |
| +17 | Unused | +47 | Unused |
| +18 | Unused | +48 | Run hour |
| +19 | Unused | +49 | Unused |
| +20 | Unused | +50 | Unused |
| +21 | Unused | +51 | Unused |
| +22 | Unused | +52 | * First variable number for roughing |
| +23 | Unused | +53 | * First variable number for finishing |
| +24 | Unused | +54 | * First variable number for chamfering |
| +25 | Unused | +55 | * First variable number of a figure block |
| +26 | Unused | +56 | * Unused |
| +27 | Unused | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Unused | +58 | * First variable number of the preceding process |
| +29 | Unused | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

**NOTE**
Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

- Y-axis Center Drilling <abailabel on Ver.2 or later>

| | | | |
|---|---|---|---|
| + 0 | Process number | +30 | Unused |
| + 1 | Unused | +31 | Unused |
| + 2 | System utilization area (∗3) | +32 | Unused |
| + 3 | Machining type (∗4) | +33 | Unused |
| + 4 | Machining area (∗5) | +34 | Unused |
| + 5 | Tool post (Spindle axis selection) (∗6) | +35 | Unused |
| + 6 | Machining cycle (∗7) | +36 | Program override |
| + 7 | Machining start point X | +37 | Unused |
| + 8 | Machining start point Z | +38 | Unused |
| + 9 | Unused | +39 | Unused |
| +10 | Unused | +40 | Skip point 1 |
| +11 | Unused | +41 | Skip point 2 |
| +12 | Unused | +42 | Skip point 3 |
| +13 | Unused | +43 | Automatic preceding process determination flag (∗12) |
| +14 | Tool number | +44 | Pass point 1 X |
| +15 | T code | +45 | Pass point 1 Z |
| +16 | Unused | +46 | Pass point 2 X |
| +17 | Unused | +47 | Pass point 2 Z |
| +18 | Tool speed (rpm) | +48 | Run hour |
| +19 | Feedrate | +49 | Unused |
| +20 | Unused | +50 | Unused |
| +21 | Unused | +51 | Unused |
| +22 | Unused | +52 | * First variable number for roughing |
| +23 | Unused | +53 | * First variable number for finishing |
| +24 | Milling gear | +54 | * First variable number for chamfering |
| +25 | Coolant (1=ON, 2=OFF) | +55 | * First variable number of a figure block |
| +26 | Unused | +56 | * Unused |
| +27 | Hole pattern (∗23) | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Nominal diameterer | +58 | * First variable number of the preceding process |
| +29 | Dwell time at hole bottom point | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

**NOTE**
Some item names depend on the version.

* : System management area –
Avoid data writing by a user program
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

● Y-axis Drilling <availabel on Ver.2 or later>

| | | | |
|---|---|---|---|
| + 0 | Process number | +30 | Machining pattern (∗14) |
| + 1 | Machining type (2) (∗13) | +31 | Decrement in depth of cut |
| + 2 | System utilization area (∗3) | +32 | Relief return amount |
| + 3 | Machining type (∗4) | +33 | Minimum value for the depth of cut |
| + 4 | Machining area (∗5) | +34 | Unused |
| + 5 | Tool post (Spindle axis selection) (∗6) | +35 | Unused |
| + 6 | Machining cycle (∗7) | +36 | Program override |
| + 7 | Machining start point X | +37 | Unused |
| + 8 | Machining start point Z | +38 | Unused |
| + 9 | Unused | +39 | Unused |
| +10 | Unused | +40 | Skip point 1 |
| +11 | Unused | +41 | Skip point 2 |
| +12 | Unused | +42 | Skip point 3 |
| +13 | Unused | +43 | Automatic preceding process determination flag (∗12) |
| +14 | Tool number | +44 | Pass point 1 X |
| +15 | T code | +45 | Pass point 1 Z |
| +16 | Unused | +46 | Pass point 2 X |
| +17 | Unused | +47 | Pass point 2 Z |
| +18 | Tool speed (rpm) | +48 | Run hour |
| +19 | Feedrate | +49 | Unused |
| +20 | Unused | +50 | Unused |
| +21 | Cutting depth | +51 | Unused |
| +22 | Unused | +52 | * First variable number for roughing |
| +23 | Unused | +53 | * First variable number for finishing |
| +24 | Milling gear | +54 | * First variable number for chamfering |
| +25 | Coolant (1=ON, 2=OFF) | +55 | * First variable number of a figure block |
| +26 | Unused | +56 | * Unused |
| +27 | Hole pattern (∗23) | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Hole diameterer | +58 | * First variable number of the preceding process |
| +29 | Dwell time at hole bottom point | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

**NOTE**
Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

● Y-axis Tapping <available on Ver.2 or later>

| | | | |
|---|---|---|---|
| + 0 | Process number | +30 | Pitch |
| + 1 | Unused | +31 | Unused |
| + 2 | System utilization area (∗3) | +32 | Unused |
| + 3 | Machining type (∗4) | +33 | Unused |
| + 4 | Machining area (∗5) | +34 | Unused |
| + 5 | Tool post (Spindle axis selection) (∗6) | +35 | Unused |
| + 6 | Machining cycle (∗7) | +36 | Unused |
| + 7 | Machining start point X | +37 | Unused |
| + 8 | Machining start point Z | +38 | Unused |
| + 9 | Unused | +39 | Unused |
| +10 | Unused | +40 | Skip point 1 |
| +11 | Unused | +41 | Skip point 2 |
| +12 | Unused | +42 | Skip point 3 |
| +13 | Unused | +43 | Automatic preceding process determination flag (∗12) |
| +14 | Tool number | +44 | Pass point 1 X |
| +15 | T code | +45 | Pass point 1 Z |
| +16 | Unused | +46 | Pass point 2 X |
| +17 | Unused | +47 | Pass point 2 Z |
| +18 | Tool speed (rpm) | +48 | Run hour |
| +19 | Feedrate | +49 | Unused |
| +20 | Unused | +50 | Unused |
| +21 | Unused | +51 | Unused |
| +22 | Unused | +52 | * First variable number for roughing |
| +23 | Unused | +53 | * First variable number for finishing |
| +24 | Milling gear | +54 | * First variable number for chamfering |
| +25 | Coolant (1=ON, 2=OFF) | +55 | * First variable number of a figure block |
| +26 | Unused | +56 | * Unused |
| +27 | Hole pattern (∗23) | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Nominal diameterer | +58 | * First variable number of the preceding process |
| +29 | Dwell time at hole bottom point | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

**NOTE**
Some item names depend on the version.

* : System management area –
Avoid data writing by a user program
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

● Y-axis Milling (rough machining) <availabel on Ver.2 or later>

| | | | |
|---|---|---|---|
| + 0 | Process number | +30 | Unused |
| + 1 | Unused | +31 | Unused |
| + 2 | System utilization area (∗3) | +32 | Shift direction (∗26) |
| + 3 | Machining type (∗4) | +33 | Unused |
| + 4 | Machining area (∗5) | +34 | Unused |
| + 5 | Tool post (Spindle axis selection) (∗6) | +35 | Unused |
| + 6 | Machining cycle (∗7) | +36 | Program override |
| + 7 | Machining start point X | +37 | Unused |
| + 8 | Machining start point Z | +38 | Unused |
| + 9 | Unused | +39 | Unused |
| +10 | Unused | +40 | Unused |
| +11 | Unused | +41 | Unused |
| +12 | Unused | +42 | Unused |
| +13 | Chamfer amount (∗15) | +43 | Unused |
| +14 | Tool number | +44 | Pass point 1 X |
| +15 | T code | +45 | Pass point 1 Z |
| +16 | Unused | +46 | Pass point 2 X |
| +17 | Unused | +47 | Pass point 2 Z |
| +18 | Tool speed (rpm) | +48 | Run hour |
| +19 | Feedrate X, Y (end face)/Y, Z (side face) | +49 | System utilization area (∗3) |
| +20 | Milling gear | +50 | Unused |
| +21 | Feedrate Z (end face)/X (side face) | +51 | Unused |
| +22 | Unused | +52 | * First variable number for roughing |
| +23 | Unused | +53 | * First variable number for finishing |
| +24 | Unused | +54 | * First variable number for chamfering |
| +25 | Coolant (1=ON, 2=OFF) | +55 | * First variable number of a figure block |
| +26 | Excape speed Z (end face)/X(side face) | +56 | * Unused |
| +27 | Cut depth | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Endmill diameter | +58 | * First variable number of the preceding process |
| +29 | Unused | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

**NOTE**

Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

● Y-axis Milling (chamfering) <available on Ver.2 or later>

| | | | |
|---|---|---|---|
| + 0 | Process number | +30 | Unused |
| + 1 | Unused | +31 | Unused |
| + 2 | System utilization area (∗3) | +32 | Shift direction (∗26) |
| + 3 | Machining type (∗4) | +33 | Unused |
| + 4 | Machining area (∗5) | +34 | Unused |
| + 5 | Tool post (Spindle axis selection) (∗6) | +35 | Unused |
| + 6 | Machining cycle (∗7) | +36 | Program override |
| + 7 | Machining start point X | +37 | Unused |
| + 8 | Machining start point Z | +38 | Unused |
| + 9 | Unused | +39 | Unused |
| +10 | Unused | +40 | Unused |
| +11 | Unused | +41 | Unused |
| +12 | Unused | +42 | Unused |
| +13 | Chamfer amount (∗15) | +43 | Unused |
| +14 | Tool number | +44 | Pass point 1 X |
| +15 | T code | +45 | Pass point 1 Z |
| +16 | Unused | +46 | Pass point 2 X |
| +17 | Unused | +47 | Pass point 2 Z |
| +18 | Tool speed (rpm) | +48 | Run hour |
| +19 | Feedrate | +49 | System utilization area (∗3) |
| +20 | Milling gear | +50 | Unused |
| +21 | Unused | +51 | Unused |
| +22 | Unused | +52 | * First variable number for roughing |
| +23 | Unused | +53 | * First variable number for finishing |
| +24 | Unused | +54 | * First variable number for chamfering |
| +25 | Coolant (1=ON, 2=OFF) | +55 | * First variable number of a figure block |
| +26 | Unused | +56 | * Unused |
| +27 | Cut depth | +57 | * Use status flag (0: Not used, 1: Used) |
| +28 | Endmil diameter | +58 | ¢* First variable number of the preceding process |
| +29 | Unused | +59 | * First variable number of the next process |

●: Ver.2 or later, ■: Ver.3 or later, (∗?): See the note below.

**NOTE**
Some item names depend on the version.

* : System management area –
Avoid data writing by a user program.
If there are no processes corresponding to +52 to +59, or if the current process corresponds to any of those processes, the area is set to null or 0.

**NOTE**

*1 : Coolant (initial setting data)
In new process generation, the data set here is automatically set as an initial value in the item of coolant for each process. If the initial setting of the item of coolant is modified during editing, the new value is reflected in all processes of the program being edited.

*2 : Finishing allowances X and Z (initial setting data)
In new process generation, the data set here is automatically set as initial values in the items of finishing allowances for each process. If the initial setting of an item of allowance is modified during editing, the new value is reflected in all processes (except C-axis notching) of the program being edited.

*3 : System area
This area is used by the system. Macro programs cannot understand any data contained in this area.

*4 : Type of machining

| | | |
|---|---|---|
| 1 : Bar machining | 2 : Tracing | 3 : Residual machining |
| 4 : End facing | 5 : Threading | 6 : Grooving |
| 7 : Necking | 8 : Center drilling | 9 : Drilling |
| 10 : Tapping | 11 : Single action | |
| 13 : Subprogram calling | 14 : C-axis center drilling | 15 : C-axis drilling |
| 16 : C-axis tapping | 17 : C-axis grooving | 18 : C-axis notching |
| 19 : C-axis cylindrical machining | 20 : Transfer | 21 : Auxiliary |
| 22 : M process | 23 : End process | 24 : Y-axis center drilling |
| 25 : Y-axis drilling | 26 : Y-axis tapping | 27 : Y-axis milling |

*5 : Machining area (depending on the type of machining)

- <Bar machining/tracing>

| | |
|---|---|
| 1 : Outer surface edge | 2 : Outer non-edge surface |
| 3 : Inner surface edge | 4 : Inner MID |
| 2 : End face edge | 6 : End face non-edge surface |
| 7 : OUT-ENDBK | 8 : OUT-MIDBK |
| 9 : INN-ENDBK | 10 : INN-MIDBK |
| 11 : FACE-BACK | 12 : FACE-MDBK (No.7 to No.12 are available only for bar machining) |

- <Residual machining>

| | | |
|---|---|---|
| 1 : Outer surface | 2 : Inner surface | 3 : End face |
| 4 : BOTTOM UED | | |

- <Threading>

| | |
|---|---|
| 1 : Outer surface | 2 : Inner surface |

- <Grooving>

| | | |
|---|---|---|
| 1 : Outer surface | 2 : Inner surface | 3 : End face |

- <Necking>

| | |
|---|---|
| 1 : Outer surface right side | 2 : Outer surface left side |
| 3 : Inner surface right side | 4 : Inner surface left side |
| 5 : End face upper side | 6 : End face lower side |

- <C-axis center drilling/C-axis drilling/C-axis tapping/C-axis grooving/C-axis notching/Y-axis center drilling/Y-axis drilling/Y-axis tapping/Y-axis milling>

| | |
|---|---|
| 1 : End face | 2 : Side face |

*6 : Tool post selection (Spindle-axis selection)

| | |
|---|---|
| 1 : Tool post 1 | 2 : Tool post 2 |
| 3 : Both turret, both spindle | |

**NOTE**

*7 : Process cycle, execution cycle
1 : Rough machining 2 : Finish machining 3 : Chamfering

Unusually, this process cycle is automatically set at new process generation to divide processes into rough machining, finish machining, and chamfering, and to arrange the order of these operation so that rough machining is performed first. However, in grooving (excluding trapezoid grooving), single action, measurement, and subprogram call, this cycle is displayed as an execution cycle on the screen to allow modifications to be made on the screen. Thus, the order can be arranged so that rough machining is performed first.

*8 : Spindle gear
1 : Automatic 2 : Low speed 3 : Intermediate speed 1
4 : Intermediate speed 2 5 : High speed

*9 : Surface speed/speed selection
0 (or null) : Surface speed 1 : Speed

*10 : Process pattern (grooving)
1 : Ordinary 2 : Slant 3 : Trapezoid
4 : Thread groove (available on and after Ver. 2)

*11 : Necking figure
1 : General-purpose 2 : Necking 1 for abrasion
3 : Necking 2 for abrasion 4 : Necking for threading

*12 : Automatic preceding process determination flag
1 : Process generated by automatic preceding process determination
0 or (null) : Process generated by ordinary editing (MDI key input). (However, if the automatic preceding process determination function is activated, this flag is set to 1 even for the last process.)

*13 : Type of machining (2)
- <Center drill/C-axis center drill> (available on Ver. 3)
  1 : Center 2 : Center+Chamfer 3 : Starting
  4 : Starting+Chamfer
- <Drilling>
  1 : Drilling 2 : Reamer 3 : Boring
  4 : Endmill (available on Ver.3) 5 : Throw-away drill (available on Ver.3)
- <C-axis drilling>
  1 : Drilling 2 : Reamer 3 : Boring
  4 : Endmill (available on Ver.3)

*14 : Machining pattern
- In case of the Ver.1 and Ver.2
  1 : Drilling 2 : Pecking 3 : Hi-spd. pecking
- In case of the Ver.3
  - <Drilling/C-axis drilling>
    1 : Hole drilling 2 : Hole pecking 3 : Hole hi-spd. pecking
    4 : Penetrate drilling 5 : Penetrate pecking
    6 : Penetrate hi-spd. pecking
  - <Endmill/C-axis endmill>
    1 : Residual cutting 2 : Spot-facing
  - <Reaming/C-axis reaming>
    1 : Hole 2 : Penetrate

*15 : Chamfering amount
When a value other than 0 is entered here, a chamfering process is generated. However, this does not apply if a chamfering processing already exists. In this case, such a process can be deleted only by pressing the [process deletion] key.

*16 : Development
These areas are used only for development display.

**NOTE**

*17 : Groove diameter/groove depth selection (Groove depth selected always when an end face is machined)
0 (or null) : Groove depth    1 : Groove diameter

*18 : Reference direction (depending on the machining area)
- <Outer surface>
  1 : Left-reference    2 : Right-reference
- <Inner surface>
  1 : Left-reference (fixed)
- <End face>
  1 : Down-reference    2 : Up-reference

*19 : Cutting method (1)
- In case of Ver.2 and Ver.3
  1 : Constant depth, half side cutting    2 : Constant depth, zigzag cutting
  3 : Constant depth, both side cutting    4 : Constant depth, half side cutting
  5 : Constant depth, zigzag cutting    6 : Constant depth, both side cutting
- In case of Ver.1
  1 : Constant depth    2 : Constant depth

*20 : Cutting method (2)
- Ver.1 only
  1 : Half side cutting    2 : Zigzag cutting

*21 : Tool post selection, process cycle
The items of tool post selection (auxiliary) and process cycle (auxiliary) are to be set with a user program.

*22 : Trapezoidal groove, thread groove - figure editing flag (initial setting)
0 : Edit end    1 : Trapezoidal groove editing
2 : Thread groove editing (available on and after Ver.2)
<Null>: No trapezoidal grooving or no thread grooving

*23: Hole pattern
1 : Circle    2 : Lattice    3 : Optional

*24: Cut number/cut depth
0 (or null) : Cut number    1 : Cut depth

*25: Thread type
1 : General    2 : Metric thread    3 : Unified thread
4 : PT thread    5 : PF thread

*26: Shift direction
- <C-axis drilling (boring)> <Ver.3 only>
  1 : +    2 : –
- <Y-axis milling>
  1 : Center    2 : Right    3 : Left

*27: Tool angle and cutting edge angle of tool to be used (necking)
The data is copied from the tool file immediately before the process is executed.

*28: End point Z/hole depth <Ver.3 only>
- <Center drill/drill/tap>
  0 (or null) : End point Z    1 : Hole depth
- <C-axis center drill/C-axis drill/C-axis tap>
  0 (or null) : Hole depth    1 : End point Z/X

*29: Type (Single action II) <Ver.3 only>
1 : Side    2 : Front drill    3 : C-open drill
4 : C-front mill    5 : C-open mill    6 : Y-open drill
7 : Y-front mill    8 : Y-open mill
(The data is used for plane selection or animated simulation in single action II.)

(c) Data structure of an auxiliary process and transfer process

The input formats and display formats of data items such as type, head, and display data depend on the data format values specified in pairs for such data items.

(Data format) = –1 : The data item is not displayed. The cursor cannot be moved to the item. The field for a data item not displayed is front-justified, that is, the field displays the next data item without leaving space.

(Data format) < 10000 : The data is numeric data, and numeric keys are used for data input. A data format value represents the number of decimal places.

(Data format) ≧ 10000 : The data provides a string indication. A soft key is used for data input.

(Data format) ≧ 20000 : The data is numeric data, and soft keys and numeric keys can be used for data input.

(Data format) = *AB** : When A= 0, the data is normally displayed.

When A=1, the data is displayed in reverse video.

When B=1, the data is displayed in red.

When B=2, the data is displayed in green.

When B=3, the data is displayed in yellow.

When B=4, the data is displayed in blue.

When B=5, the data is displayed in purple.

When B=6, the data is displayed in light blue.

When B=0, 7, or 9, the data is displayed in white.

(Data format) = 100000 : The data represents a surface speed/speed. Its display and data input processing are performed exactly in the same way as for other processes.

(Data format) = 100001 : The data provides a coolant ON/OFF indication (1=ON, 2=OFF). Soft keys and numeric keys can be used for data input.

(Data format) = 200000 : The data provides a string indication. Soft keys and numeric keys can be used for data input.

(Data format) = 300001 : The data represents a tool ID number. Soft keys and numeric keys can be used for data input.

(Data format) = 300002 : The data represents a T code. Soft keys and numeric keys can be used for data input.

(d) Examples of command format

(i) When a process is to be searched for using the process data and offset

```
#9895 = (search mode)     ; –1.
#9892 = (process data)    ; –2.
#9893 = (offset number)   ; –3.
#9892 = (process data)    ; –4.
#9893 = (offset number)   ; –5.
#9899 = –1                ; –6.
#????? = #9898            ; –7.
```

1. Sets #9895=0 when an editing program is searched for, or sets #9895 = 1 when an executable program is searched for.

2., 4. Sets reference data used for process search operation in #9892.

3., 5. Sets the offset number in #9893.

6. Enters the completion code.

7. Reads the first macro variable number, set in #9898, of a found process.

**NOTE**

1 Be sure to specify #9892 first, then specify #9893. A pair of these two variables makes up a command. Up to five pairs can be specified until –1 is set in #9899. If more than five pairs are specified, those pairs beyond the first five are ignored. If the number of specified #9892 variables does not match the number of specified #9893 variables, the error code (99) is set in #9898.

2 Process search operation starts with the process immediately after the process set in #9898. So, in order to perform process search operation starting with the first process, once perform a search operation that causes an error intentionally to set #9898=99.

(ii) When the process data of the process currently displayed is to be read

#????? = # [#9890 + (offset number)] –1.

1. Reads the offset added to #9890 containing the first macro variable number of the process currently displayed.

(iii) When the initial setting data of the program currently displayed is to be read

#????? = # [#9891 + (offset number)]  –1.

1. Reads the offset added to #9891 containing the first macro variable number of the initial setting data of the program currently displayed.

(iv) Examples of execution

- When a process being edited is to be searched for using the process data and its offset

  [A process of bar outer surface rough machining is searched for which uses a tool with the tool ID number 101.]

| | | |
|---|---|---|
| #9895 = 0 | ; | Search mode (editing program) |
| #9899 = –1 | ; | Completion code (to set #9898 = 99) |
| #9892 = 1 | ; | Bar machining |
| #9893 = 3 | ; | Type of machining |
| #9892 = 1 | ; | Rough machining |
| #9893 = 6 | ; | Machining cycle |
| #9892 = 101 | ; | Tool ID number 101 |
| #9893 = 14 | ; | Tool ID number |
| #9899 = –1 | ; | Completion code |
| #10500 = #9898 | ; | Loads the first macro variable number of a found process into #10500 |

- When the process data of the process currently displayed is to be read

  [The machining type of the process currently displayed is read.]

  #12345 = # [#9890 + 3]

  The internal code representing the machining type of the process currently displayed is read into #12345.

## E.6.3 Software Package Associated with the Interface Between User Programs and the System

### E.6.3.1 Overview

With the FANUC Super CAP T Ver. 1, 2, 3, a user program for use with the macro executor can create a dedicated screen as part of the conversational programming menu. In addition, the status of the system can be obtained with a user program.

### E.6.3.2 Interface Between the Conversational Programming Menu and User Programs

(1) System variables for specifying multi-window display

#9111 = X coordinate of the top-left corner of a window

#9112 = Y coordinate of the top-left corner of a window

#9113 = Number of characters in the X direction (horizontal direction on the screen)

#9114 = Number of characters in the Y direction (vertical direction on the screen)

#9115 = Line type of the frame (0: thin, 1: heavy)

#9116 = Color of the frame (0: black, 1: red, 2: green, 3: yellow, 4: blue, 5: purple, 6: light blue, 7: white)

#9120 = Display request to the system

– When 1 is entered, the system opens the window specified by the information of #9111 to #9116. When 2 is entered, the system closes the window. Upon completion of processing, the system initializes #9120 to 0.
To return from the detail screen to the process screen, set 99. Then the system displays the process screen.

> **NOTE**
> When any of the variables above assumes a null or a value beyond a specifiable range, the window is not displayed.

#9119 = Redisplay request to the user program

– When there is no request from the user program to close the window, the system may close the window, for example, to switch to another screen. If the system closes the window for its reason, the user program is expected to make another request to open the window. In such a case, the system sets 1. In response to this, the user program is to set the variable in the system variable to direct window display, then the user program is to output the display request to the system. When the user program is read, the system initializes #9119 to 0.
The system sets #9119 to 1 when the cursor is moved.

(2) Program numbers of user programs for screen display

O1000 = User program for detail data screen display

→ When the [DETAIL DATA] soft key is pressed (to display the detail data screen) in a transfer process, the program is called in each cycle in task processing internal to the CNC.

O1001 = User program for process data screen display

→ When a window is to be displayed on the process data screen in a transfer process, the program is called in each cycle in task processing internal to the CNC.

O1002 = Program for process data initialization

→ When a transfer process is newly generated, the program is called just once before the screen for the process is displayed.

O1003 = Program for termination processing of process data, and so forth

→ When a transfer process is terminated, the program is called just once.

O1004 = User program for detail data screen display

→ When the [DETAIL DATA] soft key is pressed (to display the detail data screen) in an auxiliary process, the program is called in each cycle in task processing internal to the CNC.

O1005 = User program for process data screen display

→ When a window is to be displayed on the process data screen in an auxiliary process, the program is called in each cycle in task processing internal to the CNC.

O1006 = Program for process data and cursor position initialization

→ When an auxiliary process is newly generated, the program is called just once before the screen for the process is displayed.

O1007 = Program for termination processing of process data, and so forth

→ When an auxiliary process is terminated, the program is called just once.

- Machining type identification

  The first variable number of the process block currently edited is set in #9890. So the type of machining can be checked by making a reference as follows:

  #????? = # [#9890 + 3]

(3) Input key monitoring and echo back display

When a screen is displayed with a user program as described in (2) above, key input can be read using system variables #8501 and #8503. The code and specification of each key is the same as in the case of a standard macro executor, with some exceptions described below.

Note, however, that the echo back display of key input need not be performed by the user program, but is performed by the conversational system as with other conversational process data screens.

Since the detail data screens for a transfer process and auxiliary process as well as the data structure are made open to user programs, the system variables listed below are added for key-in buffer display.

#9124 : In key-in buffer display on a detail data screen, this variable indicates whether the input is numeric key input or soft key input. In the latter case, this variable also indicates the number of decimal places of numeric data.

- 0 - 7 : Number of decimal places in the case of numeric key input
- 99 : Soft key input

When a value other than the above is entered, 3 is set in the case of mm input, or 4 is set in the case of inch input.

#9125 : 1 – Warning being displayed
0 – Normal state
The system may display a warning (associated with numeric calculation input, etc.) on line 20. In such a case, #9125=1 is set. When a user program provides a string indication on line 20 and below, #9125=1 is to be set in the user program. When the value of #9125 changes from 1 to 0 on a process data screen, the system rewrites line 20 and below. On a detail data screen, the key-in buffer frame only is rewritten.

#9126 : 1 – Cancel state
0 – Normal state
If the key-in buffer contains no data, #9126=1 is set if the [CAN] key is pressed followed by the [INPUT] key. #9126=0 is set when #8501 $\neq$ 0 is read, that is, when a key other than the numeric keys is pressed, and the key input is read by the user program.

Exceptions:

1. With a standard macro executor, input key monitoring is performed using #8501 to #8504. On the other hand, #8501 and #8503 only are used in this case. Any other variables including #8502 are invalid.
2. Numeric data set in #8503 is valid only in the case of [INPUT] key input (#8501=8).
3. With an addition of the numeric calculation input function, the results of calculation only are output to a user program. For this reason, [INPUT] key input for intermediate calculation and intermediate results are not output to a user program.

(4) System variables for cursor position teaching

(a) System variables for cursor position teaching

#9110 = Cursor position on process data

#9110 = 1 : Offset + 7 (type)
#9110 = 2 : Offset + 9 (head)
#9110 = 3 : Offset + 11 (display data 1)
#9110 = 4 : Offset + 13 (display data 2)
#9110 = 5 : Offset + 15 (display data 3)
#9110 = 6 : Offset + 17 (display data 4)
#9110 = 7 : Offset + 19 (display data 5)
#9110 = 8 : Offset + 21 (display data 6)
#9110 = 9 : Offset + 23 (display data 7)
#9110 = 10 : Offset + 25 (display data 8)
#9110 = 11 : Offset + 27 (display data 9)
#9110 = 12 : Offset + 29 (display data 10)
#9110 = 13 : Offset + 31 (display data 11)
#9110 = 14 : Offset + 33 (display data 12)

**NOTE**

1 Only the cursor position for a transfer process or auxiliary process is readable/writable.
2 The cursor position is not related to the position on the screen, but corresponds to the position (offset) of internal data.
3 The cursor must be entirely controlled by a user program. The system monitors #9110 at all times, and places the cursor at the screen position corresponding to the data position of the process.
4 When #9110 has been changed, the sequence must always exit from the user program. The system changes display after user program termination.

#9108 = X coordinate of the cursor position on a process data screen (read only)
#9109 = Y coordinate of the cursor position on a process data screen (read only)

On a process data screen, the current cursor position (reverse display position) is indicated using character coordinates. This can be used for window display position control.

**NOTE**

The cursor cannot be controlled using #9108 and #9109. Use #9110 to control the cursor.

(b) Communication associated with the position of the cursor between a user program and the system

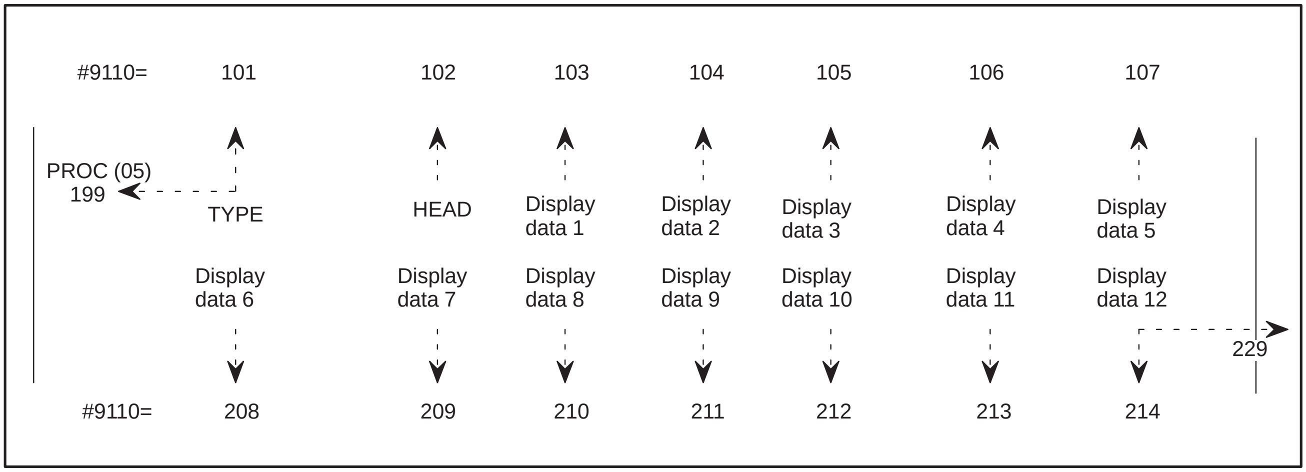
1. Cursor movement in a transfer or auxiliary process must be controlled by setting a value in #9110 with a user program. Depending on the value in #9110, the system displays the cursor at the corresponding position on the screen.

2. Any cursor movement operation in a process other than transfer and auxiliary processes is controlled by the system. Currently, the user program is not activated at that time.

3. Set, in #9110, 100 added to the current cursor position when the cursor moves outward from within a transfer or auxiliary process because the cursor key [↓] is pressed. Set, in #9110, 200 added to the current cursor position when the cursor moves outward from within a transfer or auxiliary process because the cursor key [↓] is pressed. Depending on the value in #9110, the system displays the cursor at the corresponding position.

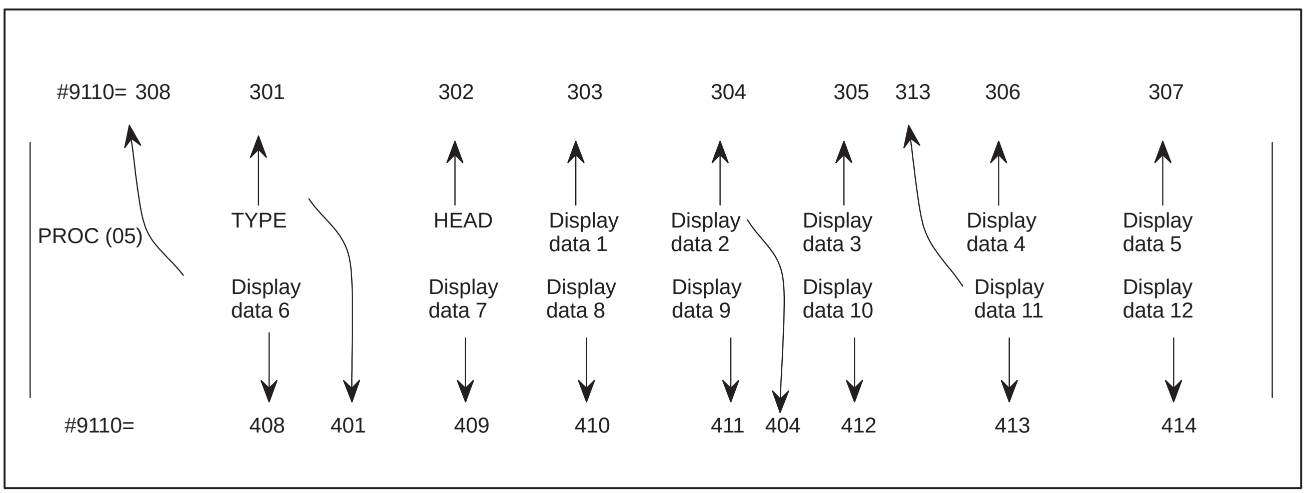
4. Set 199 in #9110 when the cursor moves to the outside of a transfer or auxiliary process from the first item (type) of the process because the cursor key [←] is pressed. Set 299 in #9110 when the cursor moves to the outside of a transfer or auxiliary process from the last item (of the process because the cursor key [→] is pressed. Depending on the value of #9110, the system displays the cursor at the corresponding position.

Example:




5. Set, in #9110, 300 added to the current cursor position when the cursor moves outward from within a transfer or auxiliary process because the page key [↑] is pressed. Set, in #9110, 400 added to the current cursor position when the cursor moves outward from within a transfer or auxiliary process because the page key [↓] is pressed. Depending on the value in #9110, the system displays the cursor at the corresponding position.

Example:




6. When the cursor moves into a transfer or auxiliary process from the outside because a page key or cursor key is pressed, the system sets the cursor position in #9110, and calls a user program.

**NOTE**

1 If an item is not displayed according to the data format in the examples above, the system front-justifies the items that follow. (That is, no space is left for an item not displayed.)

2 According to the data format of internal data, the system calculates the position of the cursor taking into any front-justification processing performed.

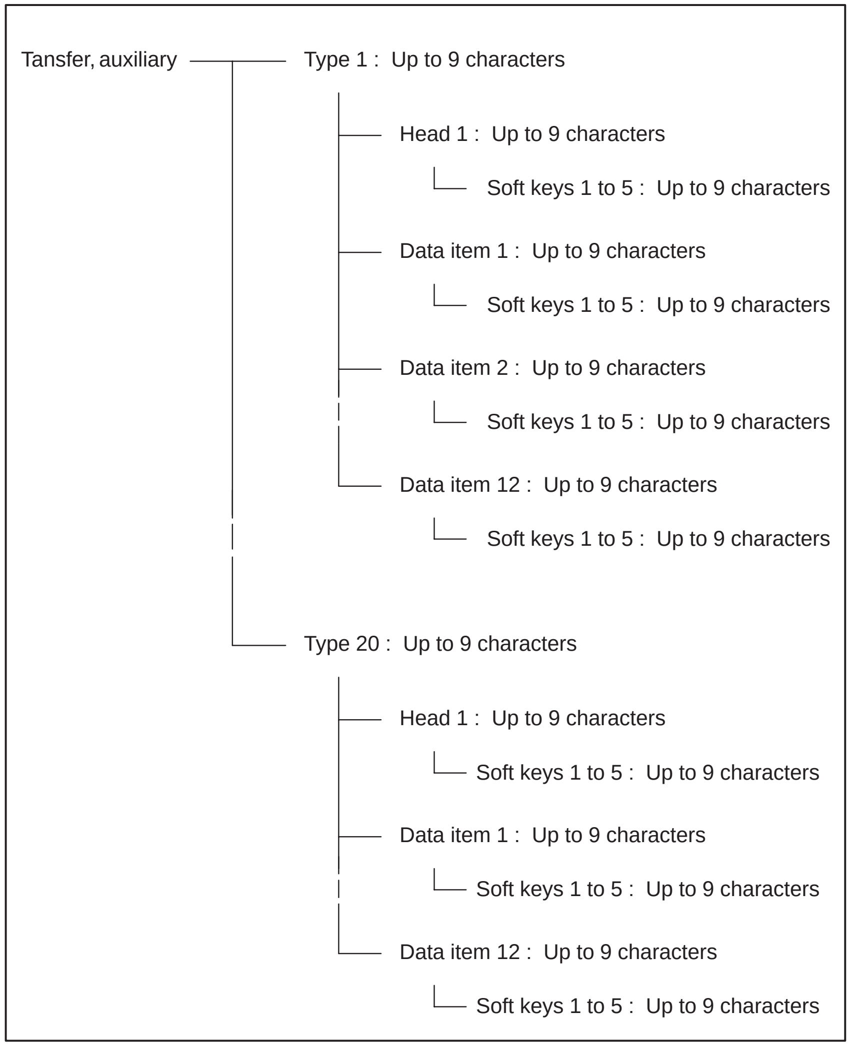
## E.6.3.3 Display Character Specification

(1) Transfer or auxiliary process display

In a transfer or or auxiliary process, an arbitrary character string can be displayed by registering display items and soft key indications in a user program.

(a) Transfer or auxiliary process details

The data items of a transfer or auxiliary process have the following structure:




With a transfer or auxiliary process, one head name and up to 12 data item names can be displayed for each of up to 20 operation types. For each data item, up to five soft keys can be displayed. These head names, data item names, and soft key names are to be registered in a user program beforehand.

**NOTE**

Up to 12 characters are potentially usable for soft key display. However, the display of only nine characters is allowed to display the same character string in a data display area.

(b) Registration of display character strings in a transfer or auxiliary process

Display character strings are registered as a user program for each display language in the following formats:

(i) Registration of type/data item display character strings

| | | | |
|---|---|---|---|
| N1101 | (H12345678) | ; | Head 1 character string |
| N1111 | (D12345678) | ; | Data item character string group 1 (12 blocks) |
| | : | | |
| N1122 | (D23456789) | ; | |
| N1201 | (H23456789) | ; | Head 2 character string |
| N1211 | (D34567890) | ; | Data item character string group 2 (12 blocks) |
| | : | | |
| N1222 | (D45678901) | ; | |
| | ≀ | | |
| N1501 | (H34567890) | ; | Head 5 character string |
| N1511 | (D56789010) | ; | Data item character string group 5 (12 blocks) |
| | : | | |
| N1522 | (D67890123) | ; | |
| | ≀ | | |
| N3001 | (H45678901) | : | Head 20 character string |
| N3011 | (D78901234) | ; | Data item character string group 20 (12 blocks) |
| | : | | |
| N3022 | (D89012345) | ; | |

**NOTE**

One head and 12 data items are displayed as a set for each type. This means that no arbitrary character string can be selected for each data item. (For details, see Section (c) (i).)

| | |
|---|---|
| O11∗1 = English | O11∗5 = Italian |
| O11∗2 = Japanese | O11∗6 = Spanish |
| O11∗3 = German | O11∗7 = Chinese (∗1) |
| O11∗4 = French | O11∗8 = Korean (∗1) |

∗ = 0 : Auxiliary process
1 : Transfer process

**NOTE**

The system displays the character string of a specified sequence number, according to the language selection parameter. English is used if no character string program specifying a language is registered.

∗1 : The Chinese and Korean languages will be supported in the future.

(ii) Registration of soft key display character strings

```
O11∗∗ ;
N0001 (S12345678) ;
N0002 (S23456789) ;
N0003 (S34567890) ;
      ≀
N5000 (S98765432) ;
```

| | |
|---|---|
| O11∗1 = English | O11∗5 = Italian |
| O11∗2 = Japanese | O11∗6 = Spanish |
| O11∗3 = German | O11∗7 = Chinese (∗1) |
| O11∗4 = French | O11∗8 = Korean (∗1) |

∗ = 2 : Auxiliary process
3 : Transfer process

> **NOTE**
> 1 The system displays the character string of a specified sequence number, according to the language selection parameter. English is used if no character string program specifying a language is registered.
> ∗1 : The Chinese and Korean languages will be supported in the future.
> 2 Up to five arbitrary soft key character strings can be selected for display. (For details, see Section (c) (ii).)

(c) Method of displaying transfer or auxiliary process display character strings

(i) Display of the display character strings of head/data item names

Display of head/data item names for a type has a tree structure. This means that when a type is selected, the character strings of one head and 12 data items are uniquely determined.

- Head/data item name display
  Set a type number (1 to 20) in the data variable for type (offset + 7). The system displays the corresponding head and data item character strings.

> **NOTE**
> The type name as an item is always “type”.

- Display of type/head/data item data
  Usually, numeric values are displayed as the data of each data item. In the case of data input using soft keys (data format ≧ 10000), however, soft key display character strings are displayed as type/head/data item data. At that time, set the sequence number of a soft key display character string in the data variable. The system displays the corresponding soft key display character string on the process screen.

**NOTE**

A value ranging from 1 to 20 can be set to specify a type. Each value corresponds to a display character string group on a one-to-one basis. So when a character string is displayed type data, the character string must be registered with a sequence number from N1 to N20 in a program for soft key display character string registration.

- Front-justified display
  If a data item is not displayed (data format=–1), the next data item is front-justified for display.
  To disable front-justification, register character strings or control the cursor as follows:
  1. Register a data item display character string consisting of blanks only beforehand.
  2. Register a soft key display character string consisting of blanks only.
  3. Set the data format of data input by soft key (data format ≧ 10000) for an item not to be displayed.
  4. Set a sequence number in the data variable of an item of (iii).
  5. Prevent the cursor from moving to an item of (iii), by using a user program.

(ii) Display of soft key display character strings

- System variables for soft key display

  #9131 : System variable representing a soft key 1 display character string
  #9132: System variable representing a soft key 2 display character string
  #9133: System variable representing a soft key 3 display character string
  #9134: System variable representing a soft key 4 display character string
  #9135: System variable representing a soft key 5 display character string

Set the sequence number of a soft key to be displayed in a variable from #9131 to 9135. The system displays the character string in the corresponding soft key position.

**NOTE**

When a variable from #9131 to #9135 has been changed, the sequence must always exit from the user program. The system changes display after user program termination.

- System variable for soft key page display

  #9136 : 1 – The system displays soft keys specified by a user program according to #9131 to #9135.
  0 – The system displays the default soft keys.

(2) Specification of display characters used with the system

(a) Use by a user program of characters used with the system

When characters are to be displayed using the macro executor, the display characters are usually specified using a display command, or a character string already registered by a user program is usually specified.

However, a very complicated user program may be required, for example, to handle many different languages.

To solve this problem, a command is added which allows character strings available in each language with the conversational system to be specified for display.

Note, however, only those characters that have preassigned codes can be specified.

G243 X (X coordinate of display start position) Y (Y coordinate of display start position) P9∗∗∗∗ ;
O1191 to O1198 ;

| | | | | |
|---|---|---|---|---|
| P90001: | “Machining program creation” | | ⋯ | N1001 |
| P90002: | “Number” | | ⋯ | N1002 |
| P90003: | “Name” | | ⋯ | N1003 |
| P90004: | “Process” | | ⋯ | N1004 |
| P90005: | “Transfer” | | ⋯ | N1005 |
| P90006: | “Head” | | ⋯ | N1006 |
| P90007: | “Tool” | | ⋯ | N1007 |
| P90008: | “Cutting” | (Soft key - 1) | ⋯ | N1008 |
| P90009: | “Condition” | (Soft key - 2) | ⋯ | N1009 |
| P90010: | “Tool” | (Soft key - 1) | ⋯ | N1010 |
| P90011: | “Data” | (Soft key - 2) | ⋯ | N1011 |
| P90012: | “Detail” | (Soft key - 1) | ⋯ | N1012 |
| P90013: | “Data” | (Soft key - 2) | ⋯ | N1013 |
| P90014: | “Product” | (Soft key - 1) | ⋯ | N1014 |
| P00015: | “Draw” | (Soft key - 2) | ⋯ | N1015 |
| P90016: | “Explanation” | (Soft key - 1) | ⋯ | N1016 |
| P90017: | “ ” | | ⋯ | N1017 |

Each string is registered in O1191 to O1198 (eight languages). P9∗∗∗∗ corresponds to each sequence number as indicated above.

(b) Title display

On the main menu screen, a title desired by each machine tool builder can be displayed. In each language (O1191 to O1198), up to 50 half-size characters can be registered in N9999(∗1).

∗1 : The Chinese and Korean languages will be supported in the future.

### E.6.3.4 Executable Macro Program Numbers

To execute a transfer or auxiliary process, the following dedicated program numbers are added:

O9018 = Program number of the transfer process executable macro
O9019 = Program number of the auxiliary process executable macro

> **NOTE**
> A program number called with G code from a user program (NC format program) is used for a transfer or auxiliary process.

### E.6.3.5 System Variable for Conversational Programming Menu Indication

#9122 = 1 : Conversational programming menu being displayed
#9122 = 0 : Different screen being displayed

> **NOTE**
> #9122 can be read also with an executable macro.

### E.6.3.6 System Variable for Window ON/OFF State Indication

#9123 = 1 : Window OFF
#9123 = 0 : Window ON

In a user program, control whether to provide window according to the value of #9123.

> **NOTE**
> Be sure to exercise cursor control at all times, regardless of the value of the system variable above.

### E.6.3.7 System Variable for Conversational Mode/NC Mode Indication

#9121 = 1 : Conversational mode
#9121 = 0 : NC mode

From the value of #9121, a user program can check whether the conversational mode or NC mode is currently selected.

> **NOTE**
> At the time of activation (including animatedsimulation), the conversational mode or NC mode is started.

#9121 can be read also with an executable macro.

### E.6.3.8 System Variable for Execution Mode Indication

#9127 = 0 : Normal state
#9127 = 1 : Simulation being executed
#9127 = 2 : NC statement being output

> **NOTE**
> #9127 can be read also with an executable macro.

### E.6.3.9 Conversational Macro Program Executed Immediately Before Execution

O1008 : Program activated when a program to be executed is selected, that is, when a soft key such as the [EXEC] and [ANIMATION] soft keys is pressed

> **NOTE**
> Only the program of the head currently selected is activated.

### E.6.3.10 Parameters and System Variables

The parameters usable for user macro programs are listed below. Each parameter corresponds to a system variable.

| Bit parameter | | | Double-word parameter | | |
|---|---|---|---|---|---|
| 9646 | —— | #9140 | 9656 | —— | #9150 |
| 9647 | —— | #9141 | 9657 | —— | #9151 |
| 9648 | —— | #9142 | 9658 | —— | #9152 |
| 9649 | —— | #9143 | 9659 | —— | #9153 |
| 9650 | —— | #9144 | 9660 | —— | #9154 |
| 9651 | —— | #9145 | 9661 | —— | #9155 |
| 9652 | —— | #9146 | 9662 | —— | #9156 |
| 9653 | —— | #9147 | 9663 | —— | #9157 |
| 9654 | —— | #9148 | 9664 | —— | #9158 |
| 9655 | —— | #9149 | 9665 | —— | #9159 |

### E.6.3.11 Program Numbers Usable with User Programs

Program numbers 2500 to 2599, 3500 to 3599, and 4500 to 4599 can be used by user programs only. Program numbers 1000 to 1199 and 9000 to 9999 are, or may be, used for the interface between the system and user macro program. User macro programs can use these program numbers provided no problem results.(*)

Other program numbers are, or will be, used by the execution macro programs of the system. User macro programs cannot, therefore, use these numbers.

> **NOTE**
> * If an interface between the system and user macro program is added in the future, user program numbers 1000 to 1199 and 9000 to 9999 may have to be changed.

## E.6.4 Data Which Can Be Referred to by Macro Programs

### E.6.4.1 Outline

In the FANUC Super CAP T/CAP II T system, macro programs corresponding to processes, such as the auxiliary and transfer processes, can refer to various data.

### E.6.4.2 Description

Program data is stored in extended P-code variables from variable No. 20000 in the Super CAP T/CAP II T system. Macro programs cannot directly refer to the program data. However, data items can be referred to indirectly. Such data items are data used in the process being executed, parameters used for conversational programming, and initial settings, which are stored in the specified areas.

(1) Program data (process data) area

Program data used in a certain process is copied to sixty macro variables, from No. 20800 to No. 20859. For details of variables used in each process, see Section 6.2.

> **NOTE**
> The same variable numbers are used for heads 1 and 2.

(2) Macro parameter area

Parameters, tool data, initial settings, and data derived from these three data items are stored in this area. These items are used in macro programs.

Initial settings 1 used in the program for head 2 are stored in sixty variables, from No. 22000 to No. 22059, immediately after the program starts. Initial settings 2 are stored in sixty variables, from No. 22060 to No. 22119, also immediately after the program starts.

> **NOTE**
> The same variable numbers are used for heads 1 and 2.

| Macro variable | Description | Processes in which the variable can be referred to | Corresponding data |
|---|---|---|---|
| #20799 | [Width of a cutter used for grooving]∗2<br>Note) Only when a cutter for grooving is registered | Grooving, transfer, and auxiliary processes | Tool data |
| #20798<br><Ver.1> | | | |
| <Ver.2 or later> | G code for calling a subprogram to rewrite an offset value in ROM when NC statements for C-axis notching are output | C-axis notching | PRM No.9779#7<br>Compile PRM No.9013 |
| #20797 | Cutting angle of a cutter<br>Tools other then round-nose tools :<br>180–([cutting edge angle]+[tools angle])<br>Round-nose tools :<br>90+[parameter setting]<br>[Angle of a tool used for threading] | Bar machining and threading | Tool data<br>PRM No. 9801 |
| #20796 | | | |
| #20795 | Flag for rough machining<br>0 : Rough machining is performed.<br>1 : Rough machining is not performed. | All processes | PRM No.9772#5 |
| #20794 | The position of the program origin is on the end face of the :<br>0 : Workpiece<br>1 : Chuck | All processes | PRM No.9786 |
| #20793 | Diameter or radius programming for the X-axis<br>0 : Radius programming is used.<br>1 : Diameter programming is used. | All processes | PRM No.1006#3 |
| #20792 | When the spindle gear is changed or the direction of the spindle rotation changes,<br>0 : code M05 is not output.<br>1 : code M05 is output. | All processes | PRM No.9772#1 |
| #20791 | The M code used for the end of a program is:<br>0 : M02<br>1 : M30<br>Note) When a program is repeated, M99 is used. | All processes | PRM No.9772#2 |
| #20790 | The machine moves from the common safety point to the point where the tool is changed along :<br>0 : One axis<br>1 : Two axes at the same time | All processes | PRM No.9772#3 |
| #20789 | Amount of return in grooving<br>[Parameter setting]∗2<br>Note) When a tool is specified | Grooving, transfer, and auxiliary processes | PRM No.9824 |
| #20788 | Limit for the depth of cut<br>[Parameter setting]∗2<br>Note) When a tool is specified | Bar machining, tracing residual machining, grooving transfer, and auxiliary processes | PRM No.9796<br>PRM No.9823 |
| #20787 | Rate of changing the depth of cut<br>[Parameter setting]/100<br>Note) If the parameter is set to 0 or 201 or more, the variable become 1.0.<br>Note) When a tool is specified | Bar machining, tracing residual machining, grooving transfer, and auxiliary processes | PRM No.9795<br>PRM No.9822 |
| #20786 | Clearance along the X-axis at the start point in grooving<br>[Parameter setting]<br>Note) When a tool is specified | Grooving, transfer, and auxiliary processes | PRM No.9820 |

| Macro variable | Description | Processes in which the variable can be referred to | Corresponding data |
|---|---|---|---|
| #20785 | Clearance along the Z-axis at the start point in grooving<br>[Parameter setting]*2<br>Note) When a tool is specified | Grooving, transfer, and auxiliary processes | PRM No.9821 |
| #20784 | Overlap between adjacent cuts in grooving<br>[Parameter setting]*[width of a tool]*2/100<br>Note) When a tool is specified | Grooving, transfer, and auxiliary processes | PRM No.9825 |
| #20783 | Dwell time at the bottom of a groove in grooving (ms)<br>Millimeter system :<br>[Parameter setting]<br>*(1000*60*3.1416*diameter) /(1000*surface speed)<br>Inch system :<br>[Parameter setting]<br>*(1000*60*3.1416*diameter) /(12*surface speed) | Grooving | PRM No.9826 |
| <Ver.2 or later> | | | |
| #20782 | Angle used for cutting with a margin from the back of a tool in intermediate bar machining<br>[Parameter setting] | Bar machining, tracing residual machining, and end facing | PRM No.9801 |
| #20781 | Clearance along the X-axis from the workpiece to the start point for cutting<br>[Parameter setting] | All processes | PRM No.9784 |
| #20780 | Clearance along the Z-axis from the workpiece to the start point for cutting<br>[Parameter setting]*2 | All processes | PRM No.9785 |
| #20779 | Clearance along the X-axis from the cut surface in machining in bites<br>[Parameter setting] | Bar machining, tracing residual machining, and end facing | PRM No.9797 |
| #20778 | Clearance along the Z-axis from the cut surface in machining in bites<br>[Parameter setting]*2 | Bar machining, tracing residual machining, and end facing | PRM No.9798 |
| #20777<br><Ver.1> | | | |
| <Ver.2 or later> | M code for turning off Y-axis mode | Y-axis proceses | PRM No.9840 |
| #20776<br><Ver.1> | | | |
| <Ver.2 or later> | M code for turning on Y-axis mode | Y-axis proceses | PRM No.9840 |
| #20775 | Cutting angle in necking<br>[Parameter setting] | Necking | PRM No.9815 |
| #20774 | Clearance along the X-axis from the line connecting the start and end points in residual machining<br>[Parameter setting] | Bar machining and residual machining | PRM No.9799 |
| #20773 | Clearance along the Z-axis from the line connecting the start and end points in residual machining<br>[Parameter setting]*2 | Bar machining and residual machining | PRM No.9800 |
| #20772 | | | |

| Macro variable | Description | Processes in which the variable can be referred to | Corresponding data |
|---|---|---|---|
| #20771 | | | |
| #20770 | Clearance along the X-axis in threading<br>[Parameter setting] | Threading | PRM No.9830 |
| #20769 | Clearance along the Z-axis in threading<br>[Parameter setting]∗2 | Threading | PRM No.9831 |
| #20768 | Minimum depth of cut in threading<br>[Parameter setting]∗2 | Threading | PRM No.9833 |
| #20767 | Maximum spindle speed with gear 1 (low speed)<br>[Parameter setting] | All processes | PRM No.3741 |
| #20766 | Maximum spindle speed with gear 2 (medium speed 1)<br>[Parameter setting] | All processes | PRM No.3742 |
| #20765 | Maximum spindle speed with gear 3 (medium speed 2)<br>[Parameter setting] | All processes | PRM No.3743 |
| #20764 | Maximum spindle speed with gear 4 (high speed)<br>[Parameter setting] | All processes | PRM No.3744 |
| #20763 | M code specifying gear 1 (low speed)<br>[Parameter setting] | All processes | PRM No.9870 |
| #20762 | M code specifying gear 2 (medium speed 1)<br>[Parameter setting] | All processes | PRM No.9871 |
| #20761 | M code specifying gear 3 (medium speed 2)<br>[Parameter setting] | All processes | PRM No.9872 |
| #20760 | M code specifying gear 4 (high speed)<br>[Parameter setting] | All processes | PRM No.9873 |
| #20759<br><Ver.1> | Common safety point along the X-axis for machining the outer surface<br>[Farthest point on the outer surface<br>(farthest point X)]+[Parameter setting] | All processes | PRM No.9780 |
| <Ver.2 or later> | Set above amount only for transfer process and for auxiliary process. | Transfer and auxiliary processes | PRM No.9780 |
| #20758<br><Ver.1> | Common safety point along the Z-axis for machining the outer surface<br>– End face of a workpiece<br>[Cutting allowance for the end face<br>(head 1/head 2)]+[Parameter setting]<br>– End face of the chuck<br>Head 1: [Length (farthest point Z)]+ [Parameter setting]<br>Head 2: [Length (farthest point Z)]– [cutting allowance for the end face (head 1)]+[Parameter setting] | All processes | PRM No.9781 |
| <Ver.2 or later> | Set above amount only for transfer process and for auxiliary process. | Transfer and auxiliary processes | PRM No.9780 |
| #20757<br><Ver.1> | Common safety point along the X-axis for machining the inner surface<br>=[Start point along the X-axis for cutting] | All processes<br>(except for sub-calling, measuring, transfer for the C-axis, and auxiliary processes for the C-axis) | Process data |
| <Ver.2 or later> | Set above amount only for transfer process and for auxiliary process. | Transfer and auxiliary processes | PRM No.9780 |

| Macro variable | Description | Processes in which the variable can be referred to | Corresponding data |
|---|---|---|---|
| #20756<br><Ver.1> | Common safety point along the Z-axis for machining the inner surface<br>– End face of a workpiece<br>[Cutting allowance for the end face (head 1/head 2)]+[Parameter setting]<br>– End face of the chuck<br>Head 1: [Length (farthest point Z)]+ [Parameter setting]<br>Head 2: [Length (farthest point Z)]– [cutting allowance for the end face (head 1)]+[Parameter setting] | All processes | PRM No.9783 |
| <Ver.2 or later> | Set above amount only for transfer process and for auxiliary process. | Transfer and auxiliary processes | PRM No.9780 |
| #20754 | Provisional offset number | All processes | |
| #20755 | | | |
| #20753 | Reference position of the chuck barrier along the X-axis : X1 | All processes | Data set previously |
| #20752 | Reference position of the chuck barrier along the Z-axis : Z1 | All processes | Data set previously |
| #20751 | Reference position of the chuck barrier along the X-axis : X2 | All processes | Data set previously |
| #20750 | Reference position of the chuck barrier along the Z-axis : Z2 | All processes | Data set previously |
| #20749 | Geometry offset for the reference tool along the X-axis | All processes | |
| #20748 | Geometry offset for the reference tool along the Z-axis | All processes | |
| #20747 | Wear offset for the reference tool along the X-axis | All processes | |
| #20746 | Wear offset for the reference tool along the Z-axis | All processes | |
| #20745 | | | |
| #20744 | Feedrate when the tool changes the direction of cutting from the normal to reverse direction in tracing<br>0 : Rapid traverse<br>1 : Cutting feed | Tracing | PRM No.9773#3 |
| #20743 | Finishing allowance in threading<br>[Parameter setting] | Threading | PRM No.9834 |
| #20742 | | | |
| #20741 | Tool orientation in pattern repeating<br>1: Right-handed, 2: Left-handed,<br>3: Positive direction,<br>4: Negative direction | Tracing | Tooling data |
| #20740 | Calling T codes are :<br>0 : Not provided<br>1 : Provided | All processes | PRM No.9773#4 |
| #20739 | Diameter or radius programming for offset<br>0 : Diameter programming is used.<br>(Only effective for axes for which diameter programming is specified.)<br>1 : Radius programming is used. | All processes | PRM No.5004#1 |
| #20738 | Upper limit of cutting feedrate | All processes | PRM No.1422 |
| #20737 | The unit used in entering data is :<br>0 : mm<br>1 : Inches | All processes | PRM No.0000#2<br>(setting) |
| #20736 | M code for preparing the tailstock | All processes | PRM No.9874 |
| #20735 | M code for storing the tailstock | All processes | PRM No.9875 |

| Macro variable | Description | Processes in which the variable can be referred to | Corresponding data |
|---|---|---|---|
| #20734 | On the graphics screen, the tailstock at the initial condition is :<br>0 : Displayed<br>1 : Not displayed | All processes | PRM No.9773#5 |
| #20733 | | | |
| #20732 | | | |
| #20731 | | | |
| #20730 | Blank drawing<br>0 : The side view is an:<br>1 : End-face view<br>2 : Exploded view | All processes | Process data |
| #20729 | Clearance at the tip of a chamfering tool<br>[Setting in the tool data]*2 | C-axis grooving, notching, and cylindrical machining | Tool data |
| #20728 | Diameter of the chamfering tool to be used<br>[Setting in the tool data] | C-axis grooving, notching, and cylindrical machining | Tool data |
| #20727 | Tool angle of the chamfering tool to be used<br>[Setting in the tool data] | C-axis grooving, notching, and cylindrical machining | Tool data |
| #20726 | Overlap at adjacent cuts in side-face notching<br>[Parameter setting]*[width of the tool] /100 | Notching (side face) | PRM No.9865<br>Tool data |
| #20725 | Width of the tool used for side cutting | Notching (side face) | Tool data |
| #20724 | M code which changes the turning mode to the milling mode | All processes | PRM No.9880 |
| #20723 | M code which changes the milling mode to the turning mode | All processes | PRM No.9881 |
| #20722 | Clearance at the start point for C-axis drilling, C-axis grooving, and C-axis cylindrical machining<br>[Parameter setting]*2 | C-axis drilling, C-axis grooving, and C-axis cylindrical machining | PRM No.9855 |
| #20721 | Maximum number registered on the graphic screen for tools used in turning<br>Setting : 1 to 16 (0 when turning tools are not registered) | All processes | Tooling data |
| #20720 | Maximum number registered on the graphic screen for tools used which rotate<br>Setting : 17 to 32 (16 when rotating tools are not registered) | All processes | Tooling data |
| #20719 | M code for stopping the rotation about the milling axis<br>[Parameter setting] (5 when a setting is 0) | All C-axis processes | PRM No.9876 |

| Macro variable | Description | Processes in which the variable can be referred to | Corresponding data |
|---|---|---|---|
| #20718 | M code for specifying the direction of rotation about the milling axis<br>– When bit 2 of parameter No. 9774 is 0, the tools to be used rotate in the :<br>[3] : Normal direction<br>[4] : Reverse direction<br>– When bit 2 of parameter No. 9774 is 1, the tools to be used rotate in the :<br>[Setting in parameter No.9877] : Normal direction<br>[Setting in parameter No.9878] : Reverse direction<br>If the settings in parameters No.9877 and No.9878 are zero or 100 or more, no value is specified. | All C-axis processes | Tool data<br>PRM No.9774#2<br>PRM No.9877<br>PRM No.9878 |
| #20717<br><Ver.1> | | | |
| <Ver.2 or later> | Limit on finishing feed amount | Bar machining, Tracing, End face, and Trapeozoid groove | Cutting condition data |
| #20716<br><Ver.1> | | | |
| <Ver.2 or later> | Limit on finishing feed amount | Automatic residual machining of bars | Cutting condition data |
| #20715 | When a program is completed, codes T0; and G28; are :<br>0 : Output<br>1 : Not output | All processes | PRM No.9774#4 |
| #20714 | When a process is completed, code M01 is :<br>0 : Not output<br>1 : Output | All processes | PRM No.9774#3 |
| #20713 | | | |
| #20712 | The optional C-axis conversational programming function is<br>0 : Provided<br>1 : Not provided | All processes | |
| #20711 | | | |
| #20710 | Geometric compensation number for the tool to be used | All processes | Process data |
| #20709 | Geometric compensation number for the tool to be used for automatic residual machining | Bar machining | Process data |
| #20708<br><Ver.1> | | | |
| <Ver.2 or later> | For the chuck/tailstock barrier function,<br>0: The second stored stroke limit is used.<br>1: The NC's chuck/tailstock barrier function is used. | All processes | Option parameter |
| #20707<br><Ver.1> | | | |
| <Ver.2 or later> | Chuck number | All processes | Initial setting |
| #20706<br><Ver.1> | | | |
| <Ver.2 or later> | Tail stock number | All processes | Initial setting |

| Macro variable | Description | Processes in which the variable can be referred to | Corresponding data |
|---|---|---|---|
| #20705 | Number registered in the graphic screen for the tool to be used<br>Note) Specify this value after address Q in the graphic command for selecting a tool. | All processes | |
| #20704 | Number registered in the graphic screen for the tool to be used for automatic residual machining.<br>Note) Specify this value after address Q in the graphic command for selecting a tool. | Bar machining | |
| #20703 | | | |
| #20702 | | | |
| #20701 | | | |
| #20700 | Point where deceleration starts in end-face machining along the X-axis<br>[Parameter setting] | Bar machining and end facing | PRM No.9807 |
| #20699 | Deceleration rate in end-face machining<br>[Parameter setting] | Bar machining and end facing | PRM No.9808 |
| #20698 | Tool geometry compensation number<br>When bit 1 of parameter No.5002 is 0, the value is the same as the wear compensation number.<br>When bit 1 of parameter No.5002 is 1, the value is the same as the tool selection number. | All C-axis processes | PRM No.5002#1 |
| #20697 | Wear compensation number of the tool to be used | All processes | Process data |
| #20696 | Wear compensation number of the tool to be used for automatic residual machining | Bar machining | Process data |
| #20695 | | | |
| #20694 | | | |
| #20693 | | | |
| #20692 | Finishing allowance in end-face machining along the Z-axis<br>[Parameter setting]*2 | All processes | Process data |
| #20691 | Selection number of the tool to be used | All processes | Process data |
| #20690 | Selection number of the tool to be used for automatic residual machining | Bar machining | Process data |
| #20689 | | | |
| #20688 | | | |
| #20687 | | | |
| #20686 | The optional interpolation function for the polar coordinate is :<br>0 : Not provided.<br>1 : Provided. | All processes | |
| #20685 | The optional cylindrical interpolation function is :<br>0 : Not provided.<br>1 : Provided. | All processes | |
| #20684 | Flag indicating NC statements are being output<br>0 : NC statements are not being output.<br>1 : NC statements are being output. | All processes | |
| #20683 | Number of the turret corresponding to the tool used in the process being performed | All processes | Process data |

| Macro variable | Description | Processes in which the variable can be referred to | Corresponding data |
|---|---|---|---|
| #20682 | Number of the turret corresponding to the tool used in the next process, or number of the turret corresponding to the tool used for automatic residual machining | All processes<br>Only in bar machining | Process data |
| #20681 | Number of the turret corresponding to the tool used in the next process | Only in bar machining | Process data |
| #20680 | Feed amount when a tool returns in reaming or boring<br>[Feed amount]∗[parameter No.9860]/10 | Reaming and boring | PRM No.9860 |
| #20679 | Surface speed of the tool at the first cut in bar machining, tracing, or end facing.<br>[Surface speed]∗[parameter No.9806]/10 | Reaming and boring | PRM No.9806 |
| #20678 | Flag indicating the change in feed amount due to the change in the cutting angle of a tool<br>0 : The feed amount is not changed.<br>1 : The feed amount is changed. | Bar machining and tracing | |
| #20677 | Feed amount for the tool with a cutting angle of 90° to 135°<br>[Feed amount]∗[Parameter setting]/100 | Bar machining and tracing | Process data<br>PRM No.9802 |
| #20676 | Feed amount for the tool with a cutting angle of 136° to 180°<br>[Feed amount]∗[Parameter setting]/100 | Bar machining and tracing | Process data<br>PRM No.9803 |
| #20675 | Feed amount for the tool with a cutting angle of 181° to 225°<br>[Feed amount]∗[Parameter setting]/100 | Bar machining and tracing | Process data<br>PRM No.9804 |
| #20674 | Feed amount for the tool with a cutting angle of 226° to 270°<br>[Feed amount]∗[Parameter setting]/100 | Bar machining and tracing | Process data<br>PRM No.9805 |
| #20673 | Clearance at the start point in boring<br>[Parameter setting] | Boring | PRM No.9859 |
| #20672 | M code for clamping the movement about the C-axis<br>[Parameter setting]<br>Note) M code for unclamping the movement about the C-axis<br>[Parameter setting]+1 | All C-axis processes | PRM No.5110 |
| #20671 | Dwell time when unclamping the C-axis is specified<br>[Parameter setting] | All C-axis processes | PRM No.5111 |
| #20670 | Clearance at the start point in C-axis tapping<br>[Parameter setting]∗2 | C-axis tapping | PRM No.9856 |
| #20669 | Switching from the turning mode to the milling mode is performed by :<br>0 : The M code<br>1 : Calling the sub-program | All processes | PRM No.9776#1 |
| #20668 | Switching from the milling mode to the turning mode is performed by :<br>0 : The M code<br>1 : Calling the sub-program | All processes | PRM No.9776#2 |
| #20667 | Outermost diameter | All processes | Initial setting |
| #20666 | Length | All processes | Initial setting |
| #20665 | Cutting allowance in end facing | All processes | Initial setting |
| #20664 | | | |
| #20663 | | | |
| #20662 | | | |

| Macro variable | Description | Processes in which the variable can be referred to | Corresponding data |
|---|---|---|---|
| #20661 | Length of a workpiece<br>Head 1 : [Length]–[cutting allowance in end facing]<br>Head 2 : [Length of the product] | All processes | Initial setting |
| #20660 | Maximum spindle speed | All processes | Initial setting |
| #20659 | Figure of a workpiece<br>1 : Bar<br>2 : Formed material | All processes | Initial setting |
| #20658 | Position on the formed material-1 : X<br>Diameter programming : [Setting]<br>Radius programming : [Setting]*2<br>Note) A value is set only when bit 7 of parameter No. 9777 is 1. | All processes | PRM No.9777#7<br>Initial setting |
| #20657 | Position on the formed material-1 : Z<br>[Setting]<br>Note) A value is set only when bit 7 of parameter No. 9777 is 1. | All processes | PRM No.9777#7<br>Initial setting |
| #20656 | Position on the formed material-2 : X<br>Diameter programming : [Setting]<br>Radius programming : [Setting]*2<br>Note) A value is set only when bit 7 of parameter No. 9777 is 1. | All processes | PRM No.9777#7<br>Initial setting |
| #20655 | Position on the formed material-2 : Z<br>[Setting]<br>Note) A value is set only when bit 7 of parameter No. 9777 is 1. | All processes | PRM No.9777#7<br>Initial setting |
| #20654 | Position on the formed material-3 : X<br>Diameter programming : [Setting]<br>Radius programming : [Setting]*2<br>Note) A value is set only when bit 7 of parameter No. 9777 is 1. | All processes | PRM No.9777#7<br>Initial setting |
| #20653 | Position on the formed material-3 : Z<br>[Setting]<br>Note) A value is set only when bit 7 of parameter No. 9777 is 1. | All processes | PRM No.9777#7<br>Initial setting |
| #20652 | Position on the formed material-4 : X<br>Diameter programming : [Setting]<br>Radius programming : [Setting]*2<br>Note) A value is set only when bit 7 of parameter No. 9777 is 1. | All processes | PRM No.9777#7<br>Initial setting |
| #20651 | Position on the formed material-4 : Z<br>[Setting]<br>Note) A value is set only when bit 7 of parameter No. 9777 is 1. | All processes | PRM No.9777#7<br>Initial setting |

| Macro variable | Description | Processes in which the variable can be referred to | Corresponding data |
|---|---|---|---|
| #20650 | Position on the formed material-5 : X<br>Diameter programming : [Setting]<br>Radius programming : [Setting]*2<br>Note) A value is set only when bit 7 of parameter No. 9777 is 1. | All processes | PRM No.9777#7<br>Initial setting |
| #20649 | Position on the formed material-5 : Z<br>[Setting]<br>Note) A value is set only when bit 7 of parameter No. 9777 is 1. | All processes | PRM No.9777#7<br>Initial setting |
| #20648 | Position on the formed material-6 : X<br>Diameter programming : [Setting]<br>Radius programming : [Setting]*2<br>Note) A value is set only when bit 7 of parameter No. 9777 is 1. | All processes | PRM No.9777#7<br>Initial setting |
| #20647 | Position on the formed material-6 : Z<br>[Setting]<br>Note) A value is set only when bit 7 of parameter No. 9777 is 1. | All processes | PRM No.9777#7<br>Initial setting |
| #20646 | Position on the formed material-7 : X<br>Diameter programming : [Setting]<br>Radius programming : [Setting]*2<br>Note) A value is set only when bit 7 of parameter No. 9777 is 1. | All processes | PRM No.9777#7<br>Initial setting |
| #20645 | Position on the formed material-7 : Z<br>[Setting]<br>Note) A value is set only when bit 7 of parameter No. 9777 is 1. | All processes | PRM No.9777#7<br>Initial setting |
| #20644 | Position on the formed material-8 : X<br>Diameter programming : [Setting]<br>Radius programming : [Setting]*2<br>Note) A value is set only when bit 7 of parameter No. 9777 is 1. | All processes | PRM No.9777#7<br>Initial setting |
| #20643 | Position on the formed material-8 : Z<br>[Setting]<br>Note) A value is set only when bit 7 of parameter No. 9777 is 1. | All processes | PRM No.9777#7<br>Initial setting |
| #20642 | Position on the formed material-9 : X<br>Diameter programming : [Setting]<br>Radius programming : [Setting]*2<br>Note) A value is set only when bit 7 of parameter No. 9777 is 1. | All processes | PRM No.9777#7<br>Initial setting |
| #20641 | Position on the formed material-9 : Z<br>[Setting]<br>Note) A value is set only when bit 7 of parameter No. 9777 is 1. | All processes | PRM No.9777#7<br>Initial setting |

| Macro variable | Description | Processes in which the variable can be referred to | Corresponding data |
|---|---|---|---|
| #20640 | Position on the formed material-10 : X<br>Diameter programming : [Setting]<br>Radius programming : [Setting]∗2<br>Note) A value is set only when bit 7 of parameter No. 9777 is 1. | All processes | PRM No.9777#7<br>Initial setting |
| #20639 | Position on the formed material-10 : Z<br>[Setting]<br>Note) A value is set only when bit 7 of parameter No. 9777 is 1. | All processes | PRM No.9777#7<br>Initial setting |
| #20638 | Position on the formed material-11 : X<br>Diameter programming : [Setting]<br>Radius programming : [Setting]∗2<br>Note) A value is set only when bit 7 of parameter No. 9777 is 1. | All processes | PRM No.9777#7<br>Initial setting |
| #20637 | Position on the formed material-11 : Z<br>[Setting]<br>Note) A value is set only when bit 7 of parameter No. 9777 is 1. | All processes | PRM No.9777#7<br>Initial setting |
| #20636 | Position on the formed material-12 : X<br>Diameter programming : [Setting]<br>Radius programming : [Setting]∗2<br>Note) A value is set only when bit 7 of parameter No. 9777 is 1. | All processes | PRM No.9777#7<br>Initial setting |
| #20635 | Position on the formed material-12 : Z<br>[Setting]<br>Note) A value is set only when bit 7 of parameter No. 9777 is 1. | All processes | PRM No.9777#7<br>Initial setting |
| #20634<br><Ver.1> | Common safety point X for turning and drilling<br>[Parameter setting] | Turning and drilling | PRM No.9787 |
| <Ver.2 or later> | | | |
| #20633<br><Ver.1> | Common safety point Z for turning and drilling<br>[Parameter setting]∗2<br>– When the program origin is positioned on the end face of the workpiece<br>[Cutting allowance on the end face]+ [Parameter setting]<br>– When the program origin is positioned on the end face of the chuck<br>Head 1: [Length (outermost point Z)] + [Parameter setting]<br>Head 2: [Length (outermost point Z)] – [cutting allow ance on the end face for head 1]+ [Parameter setting] | Turning and drilling | PRM No.9788<br>Initial setting |
| <Ver.2 or later> | | | |
| #20632 | Flag indicating spindle positioning<br>0 : Related to C-axis contour control<br>1 : Not related to C-axis contour control | All processes | PRM No.9930#1 |

| Macro variable | Description | Processes in which the variable can be referred to | Corresponding data |
|---|---|---|---|
| #20631 | The program to be called is a :<br>0 : User program stored in the NC program area<br>1 : P-code program | Sub-calling, transfer, and auxiliary processes | PRM No.9777 #1-#5<br>PRM No.9778 #0, #1 |
| #20630 | The tool post to be selected is :<br>0 : Head 1<br>1 : Head 2 | All processes | |
| #20629 | | | |
| #20628 | | | |
| #20627 | | | |
| #20626 | | | |
| #20625 | | | |
| #20624 | Cutting allowance on the end face at head 2<br>[Length]–([cutting allowance on the end face]+ [length of the product])<br>If the result is negative, set this to 0. | All processes | Initial setting |
| #20623 | Compensation along the Z-axis for copying blanks<br>When the program origin is positioned on the end face of a workpiece : A+B+C<br>When the program origin is positioned on the end face of a chuck : A+B–C<br>Where,<br>A : Z coordinate of the center of the screen at head 1<br>B : Z coordinate of the center of the screen at head 2<br>C : Length of the product | | |
| #20622 | | | |
| #20621 | | | |
| #20620 | | | |
| #20619 | Flag related to cutting on the end faces<br>0 : End facing is not performed in bar machining.<br>1 : End facing is also performed in bar machining. | Bar machining | |
| #20618 | The program to be called for switching turning to milling is a :<br>0 : User program stored in the NC program area<br>1 : P-code program | All processes | PRM No.9778#3 |
| #20617 | The program to be called for switching milling to turning is a :<br>0 : User program stored in the NC program area<br>1 : P-code program | All processes | PRM No.9778#4 |
| #20616 | Innermost diameter | All processes | Initial setting |
| #20615 | M code for calling a sub-program | All processes | Compile PRM No. 9033 |
| #20614 | Destination for calling a T code<br>0 : NC program in the RAM<br>1 : P-code program in the ROM | All processes | PRM No.9778#7 |
| #20613 | Sub-program calling by the end M code (P code)<br>0 : The end M code is output.<br>1 : The sub-program is called by the end M code. | All processes | PRM No.9779#2 |

| Macro variable | Description | Processes in which the variable can be referred to | Corresponding data |
|---|---|---|---|
| #20612 | When the direction of rotation about the milling axis is changed or the spindle gear is changed, the M code for stopping the movement about the axis is :<br>0 : Not output.<br>1 : Output. | All C-axis processes | PRM No.9774#1 |
| #20611 | M code for rotation about the milling axis in the opposite direction to that specified | C-axid drilling | PRM No.9779#0<br>1<br>PRM No.9861 |
| #20610<br><Ver.1> | | | |
| <Ver.2 or later> | Subspindle classification<br>0: Without T-series subspindle<br>1: With T-series subspindle | All processes | PRM No.9779#0<br>1<br>PRM No.9861 |
| #20609<br><Ver.1> | | | |
| <Ver.2 or later> | Initial setting data [PRODUCT LENGTH], [SETTING VALUE] | All processes | Initial setting |
| #20608<br><Ver.1> | | | |
| <Ver.2 or later> | Amount of Z coordinate shift for animated simulation of head 2 of T-series subspindle<br><Programmed zero point><br>– Workpiece end face:<br>[PRODUCT LENGTH](*2)<br>– Chuck end face:<br>[LENGTH - END REMOVAL](*2) | All processes | Initial setting |
| #20607 | Code to be used to specify turning tapping<br>0: G84, 1: G32 | Drilling | PRM No.9779#6 |
| #20606<br><Ver.1> | | | |
| <Ver.2 or later> | M code for turning on chamfering during threading | Threading | PRM No.9836 |
| #20605<br><Ver.1> | | | |
| <Ver.2 or later> | M code for turning off chamfering during threading | Threading | PRM No.9837 |
| #20604<br><Ver.1> | | | |
| <Ver.2 or later> | Specification in 96 subdivisions:<br>0: Disabled, 1: Enabled<br>(Note) 1 can be set only for a system having a T-series subspindle. | C-axis drilling and grooving | PRM No.9771#4<br>5 |
| #20603<br><Ver.1> | | | |
| <Ver.2 or later> | M code for releasing specification in 96 subdivisions.<br>(Note) 1 can be set only for a system having a T-series subspindle. | C-axis drilling and grooving | PRM No.9862 |

| Macro variable | Description | Processes in which the variable can be referred to | Corresponding data |
|---|---|---|---|
| #20602 <Ver.1> | | | |
| <Ver.2 or later> | M code for calling O9029: Set value | C-axis drilling and grooving | Compile parameter No.9032 |
| #20601 <Ver.1> | | | |
| <Ver.2 or later> | M code for calling O9028: Set value | All processes | Compile parameter No.9031 |
| #20600 <Ver.1> | | | |
| <Ver.2 or later> | M code for calling O9027: Set value | All processes | Compile parameter No.9030 |

## E.6.5 Displaying the Basic Menu Screen by a User Program (Not Supported by Ver.1.)

### E.6.5.1 Overview

FANUC Super CAP T/CAP II T allow a user-created program to display the user's own basic menu screen instead of the standard FANUC basic menu screen.

### E.6.5.2 Setting a parameter

| Parameter | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 9779 | | | USRMNU | | | | | |

bit5 (USRMNU) 0 : Standard FANUC basic menu screen is displayed.
1 : The user program for displaying the user's own basic menu screen is called.

### E.6.5.3 Details

When the USRMNU bit (bit 5 of parameter 9779) is set to 1, the user-created program for displaying the user's own basic menu screen is called.

(1) Number of the program for displaying the user's own basic menu screen, and starting that program

O1020 is assigned to the program to display basic menu screen 1, while O1021 is assigned to that to display basic menu screen 2. Each program is called only once when the screen is displayed.

(2) Screen configuration and notes

```
<Screen 1>        1          2          3          4          5          6          7
    0123456789 0123456789 0123456789 0123456789 0123456789 0123456789 0123456789 0123456789
20
21                                                                          12:34:56 EDIT
22
23
24  |   | END | 1 | 2 | 3 | 4 |   | 5 | 6 | 7 | 8 | 9 |   |

<Screen 2>        1          2          3          4          5          6          7
    0123456789 0123456789 0123456789 0123456789 0123456789 0123456789 0123456789 0123456789
20
21                                                                          12:34:56 EDIT
22
23
24  |   | END | 10 | 12 | 13 | 14 |   | 15 | 16 | 17 | 18 | 19 |   |
```

The system displays the soft keys, time, and mode as shown in the figure above. The remainder of the screen can be used freely by the user as the display area. The soft keys are controlled by the system. The user program can manage the display only. For details of the menu displayed during background editing, refer to the description of the following variable:

#8526 1 : Background editing is in progress.
0 : Background editing has stopped.

## E.6.6 Display of a Window by a User Program, Depending on the Cursor Position (Not Supported by Ver.1.)

### E.6.6.1 Overview

FANUC Super CAP T/CAP II T allow a user-created program to display the user's own window on the program process data screen. The user program determines the current position of the cursor by reading a system variable.

### E.6.6.2 Details

(1) Displaying a window, depending on the cursor position

(a) Number of the user program which displays a window

Number O1022 is assigned which the program which displays a window, depending on the cursor position.

(b) Starting the user program

User program O1022 is called:

- When the cursor is positioned to an item
- If the system issues a warning as data is being entered
- When the [▷] soft key is pressed
- When the screen is switched to the program editing screen
- When the product figure drawing window, M-detail window, or any other window opened by the system is closed
- When the [LIST ↓ ] soft key is pressed while the cursor is positioned to the program number or name

**NOTE**

1 Even in the window off state, that is, when system variable #9123 is set to 1, user program O1022 is called. In such a case, however, this program is not used to display a window. Should this be attempted in the window off state, the window may not be erased and will remain on the screen.

2 Even if the cursor is positioned to an item other than MACHINING in an auxiliary or transfer process, O1022 is called. In this case, however, O1022 must not be used to display a window. The O1022 must be used only to set system variable #9029 for displaying a window by pressing a soft key. Should O1022 be used to display a window in this state, the window may not be erased and will remain on the screen.
To display a window associated with an auxiliary or transfer process, use any of user programs O1000 to O1007.

3 If the cursor is positioned to an item other than MACHINING in an auxiliary or transfer process, the system variable associated with this function becomes undefined.

(2) System variables indicating the position of the cursor

System variables #9200 to #9203, #9108, and #9109 indicate the position of the cursor for the function for displaying a window.

(a) System variable indicating an item: #9200

In system variable #9200, a value indicating the item to which the cursor is positioned is set. The set value varies with the item, as shown below. If the cursor is positioned to the line of contour data for bar machining, pattern repeating, C-axis notching, or C-axis cylindrical machining, or to the line of single action figure data, the values of items start from 0 on each line.

<<Values of items set in system variable #9200 (italicized)>>

- Program number or name

Number=*0* Name=*1*

- Initial setting (two-spindle type)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| INITAL SET | MATERIAL | SHAPE | OUT-DIA | IN-DIA | WORK-LNG | MAX-S | COOLANT |
| | *0* | *1* | *2* | *3* | *4* | *5* | *6* |
| | FINISHX | FINISHZ | E-REMOVAL | PROD-LNG | | | |
| | *7* | *8* | *9* | *10* | | | |

- Initial setting (one-spindle type)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| INITAL SET | MATERIAL | SHAPE | OUT-DIA | IN-DIA | WORK-LNG | MAX-S | COOLANT |
| | *0* | *1* | *2* | *3* | *4* | *5* | *6* |
| | FINISHX | FINISHZ | E-REMOVAL | | | | |
| | *7* | *8* | *9* | | | | |

- Workpiece point data (point on outer surface)

| | | | | | |
|---|---|---|---|---|---|
| OUT-P X1 | Z1 | X2 | Z2 | X3 | Z3 |
| *0* | *1* | *2* | *3* | *4* | *5* |
| X4 | Z4 | X5 | Z5 | X6 | Z6 |
| *6* | *7* | *8* | *9* | *10* | *11* |
| X7 | Z7 | X8 | Z8 | X9 | Z9 |
| *12* | *13* | *14* | *15* | *16* | *17* |
| X10 | Z10 | X11 | Z11 | X12 | Z12 |
| *18* | *19* | *20* | *21* | *22* | *23* |

- Workpiece point data (point on inner surface)

| | | | | | |
|---|---|---|---|---|---|
| IN-P X1 | Z1 | X2 | Z2 | X3 | Z3 |
| *0* | *1* | *2* | *3* | *4* | *5* |
| X4 | Z4 | X5 | Z5 | X6 | Z6 |
| *6* | *7* | *8* | *9* | *10* | *11* |
| X7 | Z7 | X8 | Z8 | X9 | Z9 |
| *12* | *13* | *14* | *15* | *16* | *17* |
| X10 | Z10 | X11 | Z11 | X12 | Z12 |
| *18* | *19* | *20* | *21* | *22* | *23* |

- PROCESS DATA : the value at every items are defined.

| Data value | Name of item | | Data value | Name of item | |
|---|---|---|---|---|---|
| *0* | MACHINING | | *50* | FEEDRATE1 | <C-Axis Cylinder> |
| *1* | MACHIN-2 | | *51* | FEEDRATE2 | <C-Axis Grooving, Y-Axis Mill> |
| *2* | AREA | | *52* | FEEDRATE2 | <C-Axis Cylinder> |
| *3* | HEAD or SPINDLE | | *53* | BEVEL-AM | |
| *4* | PATTERN | | *54* | STAT-PZ | |
| *5* | BASIS | | *55* | END-PTZ | |
| *6* | SHAPE | <Necking > | *56* | FINISHX | |
| *7* | FEEDRATE | | *57* | FINISHZ | |
| *8* | PROGRAM | | *58* | STAT-PX | |
| *9* | INTRVAL | | *59* | DEPTH | |
| *10* | SHAPE | | *60* | M | |
| *11* | RETURN | | *61* | RETURN-X | |
| *12* | END-M | | *62* | RETURN-Z | |
| *13* | ROUGHNES | | *63* | RETURN-C | |
| *14* | TOOL-NO | | *64* | LOOP | |
| *15* | CUT-SPD or REV/MIN | | *65* | CUT-METHD | <Threading> |
| *16* | FEED/REV | | *66* | THRD-TYPE | |
| *17* | CUT DEPTH/CUT NUMBER | <Threading> | *67* | THRD ANGL | |
| *18* | T-CODE | | *68* | THRD CNT | |
| *19* | ROUGHNES | <BAR: Auto Res. Cut> | *69* | HOLE-PTN | |
| *20* | TOOL-NO | <BAR: Auto Res. Cut> | *70* | SHIFT-DR | |
| *21* | CUT-SPD or REV/MIN | <BAR: Auto Res. Cut> | *71* | STAT-PX | |
| *22* | FEED/REV | <BAR: Auto Res. Cut> | *72* | STAT-PZ | |
| *23* | CUT DEPTH | <BAR: Auto Res. Cut> | *73* | FEEDRATE1 | <Y-Axis Milling: Face> |
| *24* | T-CODE | <BAR: Auto Res. Cut> | *74* | FEEDRATE1 | <Y-Axis Milling: Side> |
| *25* | REMOVALX | | *75* | ESCAPE-SZ | |
| *26* | REMOVALZ | | *76* | ESCAPE-SX | |
| *27* | LEAD | | *77* | RESID-PZ | |
| *28* | CHAMFER | | *78* | CHAMF-LNG | |
| *29* | WIDTH | | *79* | STAT-PZ | |
| *30* | CUT-DPTH | | *80* | CHAMF-DIA | |
| *31* | DWELL | | *81* | TYPE | <Single Action II> |
| *32* | ANGLE | | *82* | DATA A | <Call Sub II. > |
| *33* | AMNT A | | *83* | DATA B | <Call Sub II. > |
| *34* | AMNT B | | *84* | DATA C | <Call Sub II. > |
| *35* | AMNT C | | *85* | DATA I | <Call Sub II. > |
| *36* | AMNT D | | *86* | DATA J | <Call Sub II. > |
| *37* | STNDRD-D | | *87* | DATA K | <Call Sub II. > |
| *38* | END-PT <Ver. 2> | \| END-PTZ/DEPTH <Ver. 3> | *88* | DATA D | <Call Sub II. > |
| *39* | NOMINL-D | | *89* | DATA E | <Call Sub II. > |
| *40* | DWELL | | *90* | DATA F | <Call Sub II. > |
| *41* | HOLE-DIA | | *91* | DATA H | <Call Sub II. > |
| *42* | SHIFT | | *92* | DATA M | <Call Sub II. > |
| *43* | PITCH | | *93* | | |
| *44* | DATA1 | | *94* | | |
| *45* | DATA2 | | *95* | | |
| *46* | DATA3 | | *96* | | |
| *47* | DATA4 | | *97* | | |
| *48* | GRV-DIA | | *98* | | |
| *49* | FEEDRATE1 | <C-Axis Grooving> | *99* | | |

- When a new process is created

| |
|---|
| PROC(01) |
| *0* |

- Bar machining (TT, T two-spindle type)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Bar machining (rough) | PROC(01) | AREA | HEAD | TOOL-NO | CUT-SPD | FEED/REV | CUT-DPTH | T-CODE |
| | *0* | *2* | *3* | *14* | *15* | *16* | *17* | *18* |
| | | AREA RESIDUAL | | TOOL-NO | CUT-SPD | FEED/REV | CUT-DPTH | T-CODE |
| | | | | *20* | *21* | *22* | *23* | *24* |
| Bar machining (finish) | PROC(01) | AREA | HEAD | ROUGHNES | TOOL-NO | CUT-SPD | FEED/REV | T-CODE |
| | *0* | *2* | *3* | *13* | *14* | *15* | *16* | *18* |
| | | AREA RESIDUAL | | ROUGHNES | TOOL-NO | CUT-SPD | FEED/REV | T-CODE |
| | | | | *19* | *20* | *21* | *22* | *24* |

- Bar machining (T one-spindle type)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Bar machining | PROC(01) | AREA | | TOOL-NO | CUT-SPD | FEED/REV | CUT-DPTH | T-CODE |
| (rough) | *0* | *2* | | *14* | *15* | *16* | *17* | *18* |
| | | AREA | | TOOL-NO | CUT-SPD | FEED/REV | CUT-DPTH | T-CODE |
| | | RESIDUAL | | 20 | 21 | 22 | 23 | 24 |
| Bar machining | PROC(01) | AREA | | ROUGHNES | TOOL-NO | CUT-SPD | FEED/REV | T-CODE |
| (finish) | *0* | *2* | | *13* | *14* | *15* | *16* | *18* |
| | | AREA | | ROUGHNES | TOOL-NO | CUT-SPD | FEED/REV | T-CODE |
| | | RESIDUAL | | *19* | *20* | *21* | *22* | *24* |

- Pattern repeating (TT, T two-spindle type)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Pattern repeating | PROC(01) | AREA | HEAD | REMOVALX | REMOVALZ | TOOL-NO | CUT-SPD | T-CODE |
| (rough) | *0* | *2* | *3* | *25* | *26* | *14* | *15* | *18* |
| | | FEED/REV | CUT-DPTH | | | | | |
| | | *16* | *17* | | | | | |
| Pattern reapting | PROC(01) | AREA | HEAD | ROUGHNES | TOOL-NO | CUT-SPD | FEED/REV | T-CODE |
| (finish) | *0* | *2* | *3* | *13* | *14* | *15* | *16* | *18* |

- Pattern repeating (T one-spindle type)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Pattern repeating | PROC(01) | AREA | | REMOVALX | REMOVALZ | TOOL-NO | CUT-SPD | T-CODE |
| (rough) | *0* | *2* | | *25* | *26* | *14* | *15* | *18* |
| | | FEED/REV | CUT-DPTH | | | | | |
| | | *16* | *17* | | | | | |
| Pattern reapting | PROC(01) | AREA | | ROUGHNES | TOOL-NO | CUT-SPD | FEED/REV | T-CODE |
| (finish) | *0* | *2* | | *13* | *14* | *15* | *16* | *18* |

- Residual machining (TT, T two-spindle type)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Residual machining | PROC(01) | AREA | HEAD | TOOL-NO | CUT-SPD | FEED/REV | CUT-DPTH | T-CODE |
| (rough) | *0* | *2* | *3* | *14* | *15* | *16* | *17* | *18* |
| Residual machining | PROC(01) | AREA | HEAD | ROUGHNES | TOOL-NO | CUT-SPD | FEED/REV | T-CODE |
| (finish) | *0* | *2* | *3* | *13* | *14* | *15* | *16* | *18* |

- Residual machining (T one-spindle type)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Residual machining | PROC(01) | AREA | | TOOL-NO | CUT-SPD | FEED/REV | CUT-DPTH | T-CODE |
| (rough) | *0* | *2* | | *14* | *15* | *16* | *17* | *18* |
| Residual machining | PROC(01) | AREA | | ROUGHNES | TOOL-NO | CUT-SPD | FEED/REV | T-CODE |
| (finish) | *0* | *2* | | *13* | *14* | *15* | *16* | *18* |

- End facing (TT, T two-spindle type)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| End facing | PROC(01) | | HEAD | TOOL-NO | CUT-SPD | FEED/REV | CUT-DPTH | T-CODE |
| (rough) | *0* | | *3* | *14* | *15* | *16* | *17* | *18* |
| End facing | PROC(01) | | HEAD | ROUGHNES | TOOL-NO | CUT-SPD | FEED/REV | T-CODE |
| (finish) | *0* | | *3* | *13* | *14* | *15* | *16* | *18* |

- End facing (T one-spindle type)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| End facing | PROC(01) | | | TOOL-NO | CUT-SPD | FEED/REV | CUT-DPTH | T-CODE |
| (rough) | *0* | | | *14* | *15* | *16* | *17* | *18* |
| End facing | PROC(01) | | | ROUGHNES | TOOL-NO | CUT-SPD | FEED/REV | T-CODE |
| (finish) | *0* | | | *13* | *14* | *15* | *16* | *18* |

* For the T two-spindle type, UNIT is replaced with SPINDLE.

- Threading (TT, T two-spindle type)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Threading | PROC(01) | AREA | HEAD | THRD TYPE | THRD ANGL | CUT-METHD | LEAD | |
| (general-purpose) | *0* | *2* | *3* | *66* | *67* | *65* | *27* | |
| | | TOOL-NO | CUT-SPD | CUT-NMBR | CHAMFER | | | T-CODE |
| | | *14* | *15* | *17* | *28* | | | *18* |
| Threading | PROC(01) | AREA | HEAD | THRD TYPE | CUT-METHD | LEAD | | |
| (metric) | *0* | *2* | *3* | *66* | *65* | *27* | | |
| | | TOOL-NO | CUT-SPD | CUT-NMBR | CHAMFER | | | T-CODE |
| | | *14* | *15* | *17* | *28* | | | *18* |
| Threading (unified PT, PF) | PROC(01) | AREA | HEAD | THRD TYPE | CUT-METHD | THRD CNT | | |
| | *0* | *2* | *3* | *66* | *65* | *68* | | |
| | | TOOL-NO | CUT-SPD | CUT-NMBR | CHAMFER | | | T-CODE |
| | | *14* | *15* | *17* | *28* | | | *18* |

- Threading (T one-spindle type)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Threading | PROC(01) | AREA | | THRD TYPE | THRD ANGL | CUT-METHD | LEAD | |
| (general-purpose) | *0* | *2* | | *66* | *67* | *65* | *27* | |
| | | TOOL-NO | CUT-SPD | CUT-NMBR | CHAMFER | | | T-CODE |
| | | *14* | *15* | *17* | *28* | | | *18* |
| Threading(metric) | PROC(01) | AREA | | THRD TYPE | CUT-METHD | LEAD | | |
| | *0* | *2* | | *66* | *65* | *27* | | |
| | | TOOL-NO | CUT-SPD | CUT-NMBR | CHAMFER | | | T-CODE |
| | | *14* | *15* | *17* | *28* | | | *18* |
| Threading | PROC(01) | AREA | | THRD TYPE | CUT-METHD | THRD CNT | | |
| (unified PT, PF) | *0* | *2* | | *66* | *65* | *68* | | |
| | | TOOL-NO | CUT-SPD | CUT-NMBR | CHAMFER | | | T-CODE |
| | | *14* | *15* | *17* | *28* | | | *18* |

- Grooving (TT, T two-spindle type)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Grooving | PROC(01) | AREA | HEAD | BASIS | PATTERN | WIDTH | TOOL-NO | T-CODE |
| <standard> | *0* | *2* | *3* | *5* | *4* | *29* | *14* | *18* |
| | | CUT-SPD | FEED/REV | CUT-DPTH | DWELL | | | |
| | | *15* | *16* | *30* | *31* | | | |
| Grooving | PROC(01) | AREA | HEAD | BASIS | PATTERN | WIDTH | ANGLE | |
| <slanted> | *0* | *2* | *3* | *5* | *4* | *29* | *32* | |
| | | TOOL-NO | CUT-SPD | FEED/REV | CUT-DPTH | DWELL | | T-CODE |
| | | *14* | *15* | *16* | *30* | *31* | | *18* |
| Grooving (rough) | PROC(01) | AREA | HEAD | BASIS | PATTERN | WIDTH | TOOL-NO | T-CODE |
| <trapezoidal> | *0* | *2* | *3* | *5* | *4* | *29* | *14* | *18* |
| | | CUT-SPD | FEED/REV | CUT-DPTH | DWELL | | | |
| | | *15* | *16* | *30* | *31* | | | |
| Grooving (finish) | PROC(01) | AREA | HEAD | BASIS | PATTERN | WIDTH | ROUGHNES | |
| <trapezoidal> | *0* | *2* | *3* | *5* | *4* | *29* | *13* | |
| | | TOOL-NO | CUT-SPD | FEED/REV | | | | T-CODE |
| | | *14* | *15* | *16* | | | | *18* |
| Grooving | PROC(01) | AREA | HEAD | BASIS | PATTERN | WIDTH | ANGLE | |
| <thread> | *0* | *2* | *3* | *5* | *4* | *29* | *32* | |
| | | TOOL-NO | CUT-SPD | FEED/REV | CUT-DPTH | DWELL | | T-CODE |
| | | *14* | *15* | *16* | *30* | *31* | | *18* |

### ● Grooving (T one-spindle type)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Grooving | PROC(01) | AREA | | BASIS | PATTERN | WIDTH | TOOL-NO | T-CODE |
| <standard> | *0* | *2* | | *5* | *4* | *29* | *14* | *18* |
| | | CUT-SPD | FEED/REV | CUT-DPTH | DWELL | | | |
| | | *15* | *16* | *30* | *31* | | | |
| Grooving | PROC(01) | AREA | | BASIS | PATTERN | WIDTH | ANGLE | |
| <slanted> | *0* | *2* | | *5* | *4* | *29* | *32* | |
| | | TOOL-NO | CUT-SPD | FEED/REV | CUT-DPTH | DWELL | | T-CODE |
| | | *14* | *15* | *16* | *30* | *31* | | *18* |
| Grooving (rough) | PROC(01) | AREA | | BASIS | PATTERN | WIDTH | TOOL-NO | T-CODE |
| <trapezoidal> | *0* | *2* | | *5* | *4* | *29* | *14* | *18* |
| | | CUT-SPD | FEED/REV | CUT-DPTH | DWELL | | | |
| | | *15* | *16* | *30* | *31* | | | |
| Grooving (finish) | PROC(01) | AREA | | BASIS | PATTERN | WIDTH | ROUGHNES | |
| <trapezoidal> | *0* | *2* | | *5* | *4* | *29* | *13* | |
| | | TOOL-NO | CUT-SPD | FEED/REV | | | | T-CODE |
| | | *14* | *15* | *16* | | | | *18* |
| Grooving | PROC(01) | AREA | | BASIS | PATTERN | WIDTH | ANGLE | |
| <thread> | *0* | *2* | | *5* | *4* | *29* | *32* | |
| | | TOOL-NO | CUT-SPD | FEED/REV | CUT-DPTH | DWELL | | T-CODE |
| | | *14* | *15* | *16* | *30* | *31* | | *18* |

### ● Necking (TT, T two-spindle type)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Necking | PROC(01) | | HEAD | AREA | SHAPE | ROUGHNES | TOOL-NO | T-CODE |
| <general-purpose> | *0* | | *3* | *2* | *6* | *13* | *14* | *18* |
| | | CUT-SPD | FEED/REV | AMNT A | AMNT B | AMNT C | AMNT D | |
| | | *15* | *16* | *33* | *34* | *35* | *36* | |
| Necking | PROC(01) | | HEAD | AREA | SHAPE | STNDRD-D | ROUGHNES | |
| <DIN> | *0* | | *3* | *2* | *6* | *37* | *13* | |
| | | TOOL-NO | CUT-SPD | FEED/REV | | | | T-CODE |
| | | *14* | *15* | *16* | | | | *18* |

### ● Necking (T one-spindle type)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Necking | PROC(01) | | | AREA | SHAPE | ROUGHNES | TOOL-NO | T-CODE |
| <general-purpose> | *0* | | | *2* | *6* | *13* | *14* | *18* |
| | | CUT-SPD | FEED/REV | AMNT A | AMNT B | AMNT C | AMNT D | |
| | | *15* | *16* | *33* | *34* | *35* | *36* | |
| Necking | PROC(01) | | | AREA | SHAPE | STNDRD-D | ROUGHNES | |
| <DIN> | *0* | | | *2* | *6* | *37* | *13* | |
| | | TOOL-NO | CUT-SPD | FEED/REV | | | | T-CODE |
| | | *14* | *15* | *16* | | | | *18* |

### ● Center drilling (TT, T two-spindle type) <Ver.2>

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Center drilling | PROC(01) | | HEAD | END-PT | NOMINL-D | TOOL-NO | CUT-SPD | T-CODE |
| | *0* | | *3* | *38* | *39* | *14* | *15* | *18* |
| | | FEED/REV | DWELL | | | | | |
| | | *16* | *40* | | | | | |

* For the T two-spindle type, UNIT is replaced with SPINDLE.

● Center drilling (TT, T two-spindle type) <Ver.3>

[Center drilling, starting]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Center drilling | PROC(01) | MACHN-2 | HEAD | HOLE-DIA | STAT-PZ | END-PTZ | TOOL-NO | T-CODE |
| | *0* | *1* | *3* | *39* | *79* | *38* | *14* | *18* |
| | | CUT-SPD | FEED/REV | DWELL | | | | |
| | | *15* | *16* | *40* | | | | |

[Center drilling + chamfering, starting + chamfering]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Center drilling | PROC(01) | MACHN-2 | HEAD | CHAMF-DIA | STAT-PZ | END-PTZ | TOOL-NO | T-CODE |
| | *0* | *1* | *3* | *80* | *79* | *38* | *14* | *18* |
| | | CUT-SPD | FEED/REV | DWELL | | | | |
| | | *15* | *16* | *40* | | | | |

● Center drilling (T one-spindle type) <Ver.2>

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Center drilling | PROC(01) | | | END-PT | NOMINL-D | TOOL-NO | CUT-SPD | T-CODE |
| | *0* | | | *38* | *39* | *14* | *15* | *18* |
| | | FEED/REV | DWELL | | | | | |
| | | *16* | *40* | | | | | |

● Center drilling (T one-spindle type) <Ver.3>

[Center drilling, starting]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Center drilling | PROC(01) | MACHN-2 | | HOLE-DIA | STAT-PZ | END-PTZ | TOOL-NO | T-CODE |
| | *0* | *1* | | *39* | *79* | *38* | *14* | *18* |
| | | CUT-SPD | FEED/REV | DWELL | | | | |
| | | *15* | *16* | *40* | | | | |

[Center drilling + chamfering, starting + chamfering]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Center drilling | PROC(01) | MACHN-2 | | CHAMF-DIA | STAT-PZ | END-PTZ | TOOL-NO | T-CODE |
| | *0* | *1* | | *80* | *79* | *38* | *14* | *18* |
| | | CUT-SPD | FEED/REV | DWELL | | | | |
| | | *15* | *16* | *40* | | | | |

● Drilling (TT, T two-spindle type) <Ver.2>

[Up to 40 tools registered]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Drilling | PROC(01) | | HEAD | END-PT | HOLE-DIA | TOOL-NO | CUT-SPD | T-CODE |
| | *0* | | *3* | *38* | *41* | *14* | *15* | *18* |
| | | FEED/REV | DWELL | PATTERN | CUT-DPTH | | | |
| | | *16* | *40* | *4* | *30* | | | |

[Up to 99 tools registered: drilling]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Drilling | PROC(01) | MACHN-2 | HEAD | END-PT | HOLE-DIA | TOOL-NO | CUT-SPD | T-CODE |
| | *0* | *1* | *3* | *38* | *41* | *14* | *15* | *18* |
| | | FEED/REV | DWELL | PATTERN | CUT-DPTH | | | |
| | | *16* | *40* | *4* | *30* | | | |

[Up to 99 tools registered: reaming]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Drilling | PROC(01) | MACHN-2 | HEAD | END-PT | HOLE-DIA | TOOL-NO | CUT-SPD | T-CODE |
| | *0* | *1* | *3* | *38* | *41* | *14* | *15* | *18* |
| | | FEED/REV | DWELL | | | | | |
| | | *16* | *40* | | | | | |

[Up to 99 tools registered: boring]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Drilling | PROC(01) | MACHN-2 | HEAD | END-PT | HOLE-DIA | TOOL-NO | CUT-SPD | T-CODE |
| | *0* | *1* | *3* | *38* | *41* | *14* | *15* | *18* |
| | | FEED/REV | DWELL | SHIFT | | | | |
| | | *16* | *40* | *42* | | | | |

* For the T two-spindle type, UNIT is replaced with SPINDLE.

- Drilling (TT, T two-spindle type) <Ver.3>

[Drilling]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Drilling | PROC(01) | MACHIN-2 | HEAD | PATTERN | PROC-DIA | STAT-PZ | END-PTZ | |
| | 0 | 1 | 3 | 4 | 41 | 79 | 38 | |
| | | TOOL-NO | CUT-SPD | FEED/REV | DWELL | CUT-DEPTH | | T-CODE |
| | | 14 | 15 | 16 | 40 | 30 | | 18 |

[Reaming]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Drilling | PROC(01) | MACHIN-2 | HEAD | PATTERN | PROC-DIA | CHAMF-LNG | | |
| | 0 | 1 | 3 | 4 | 41 | 78 | | |
| | | STAT-PZ | END-PTZ | TOOL-NO | CUT-SPD | FEED-REV | DWELL | T-CODE |
| | | 79 | 38 | 14 | 15 | 16 | 40 | 18 |

[Boring]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Drilling | PROC(01) | MACHIN-2 | HEAD | PROC-DIA | STAT-PZ | END-PTZ | TOOL-NO | T-CODE |
| | 0 | 1 | 3 | 41 | 79 | 38 | 14 | 18 |
| | | CUT-SPD | FEED/REV | DWELL | SHIFT | | | |
| | | 15 | 16 | 40 | 42 | | | |

[End milling]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Drilling | PROC(01) | MACHIN-2 | HEAD | PATTERN | PROC-DIA | STAT-PZ | RESID-PZ | |
| | 0 | 1 | 3 | 4 | 41 | 79 | 77 | |
| | | END-PTZ | TOOL-NO | CUT-SPD | FEED/REV1 | DWELL | | T-CODE |
| | | 38 | 14 | 15 | 16 | 40 | | 18 |

[Throw-away drilling]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Drilling | PROC(01) | MACHIN-2 | HEAD | PATTERN | PROC-DIA | STAT-PZ | END-PTZ | |
| | 0 | 1 | 3 | 4 | 41 | 79 | 38 | |
| | | TOOL-NO | CUT-SPD | FEED/REV | DWELL | CUT-DEPTH | | T-CODE |
| | | 14 | 15 | 16 | 40 | 30 | | 18 |

- Drilling (T one-spindle type) <Ver.2>

[Up to 40 tools registered]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Drilling | PROC(01) | | | END-PT | HOLE-DIA | TOOL-NO | CUT-SPD | T-CODE |
| | 0 | | | 38 | 41 | 14 | 15 | 18 |
| | | FEED/REV | DWELL | PATTERN | CUT-DPTH | | | |
| | | 16 | 40 | 4 | 30 | | | |

[Up to 99 tools registered: drilling]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Drilling | PROC(01) | MACHN-2 | | END-PT | HOLE-DIA | TOOL-NO | CUT-SPD | T-CODE |
| | 0 | 1 | | 38 | 41 | 14 | 15 | 18 |
| | | FEED/REV | DWELL | PATTERN | CUT-DPTH | | | |
| | | 16 | 40 | 4 | 30 | | | |

[Up to 99 tools registered: reaming]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Drilling | PROC(01) | MACHN-2 | | END-PT | HOLE-DIA | TOOL-NO | CUT-SPD | T-CODE |
| | 0 | 1 | | 38 | 41 | 14 | 15 | 18 |
| | | FEED/REV | DWELL | | | | | |
| | | 16 | 40 | | | | | |

[Up to 99 tools registered: boring]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Drilling | PROC(01) | MACHN-2 | | END-PT | HOLE-DIA | TOOL-NO | CUT-SPD | T-CODE |
| | 0 | 1 | | 38 | 41 | 14 | 15 | 18 |
| | | FEED/REV | DWELL | SHIFT | | | | |
| | | 16 | 40 | 42 | | | | |

* For the T two-spindle type, UNIT is replaced with SPINDLE.

- Drilling (T one-spindle type) <Ver.3>

[Drilling]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Drilling | PROC(01) | MACHIN-2 | | PATTERN | PROC-DIA | STAT-PZ | END-PTZ | |
| | 0 | 1 | | 4 | 41 | 79 | 38 | |
| | | TOOL-NO | CUT-SPD | FEED/REV | DWELL | CUT-DEPTH | | T-CODE |
| | | 14 | 15 | 16 | 40 | 30 | | 18 |

[Reaming]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Drilling | PROC(01) | MACHIN-2 | | PATTERN | PROC-DIA | CHAMF-LNG | | |
| | 0 | 1 | | 4 | 41 | 78 | | |
| | | STAT-PZ | END-PTZ | TOOL-NO | CUT-SPD | FEED-REV | DWELL | T-CODE |
| | | 79 | 38 | 14 | 15 | 16 | 40 | 18 |

[Boring]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Drilling | PROC(01) | MACHIN-2 | | PROC-DIA | STAT-PZ | END-PTZ | TOOL-NO | T-CODE |
| | 0 | 1 | | 41 | 79 | 38 | 14 | 18 |
| | | CUT-SPD | FEED/REV | DWELL | SHIFT | | | |
| | | 15 | 16 | 40 | 42 | | | |

[End milling]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Drilling | PROC(01) | MACHIN-2 | | PATTERN | PROC-DIA | STAT-PZ | RESID-PZ | |
| | 0 | 1 | | 4 | 41 | 79 | 77 | |
| | | END-PTZ | TOOL-NO | CUT-SPD | FEED/REV1 | DWELL | | T-CODE |
| | | 38 | 14 | 15 | 16 | 40 | | 18 |

[Throw-away drilling]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Drilling | PROC(01) | MACHIN-2 | | PATTERN | PROC-DIA | STAT-PZ | END-PTZ | |
| | 0 | 1 | | 4 | 41 | 79 | 38 | |
| | | TOOL-NO | CUT-SPD | FEED/REV | DWELL | CUT-DEPTH | | T-CODE |
| | | 14 | 15 | 16 | 40 | 30 | | 18 |

- Tapping (TT, T two-spindle type) <Ver. 2>

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Tapping | PROC(01) | | HEAD | END-PT | NOMINL-D | PITCH | TOOL-NO | T-CODE |
| | 0 | | 3 | 38 | 39 | 43 | 14 | 18 |
| | | CUT-SPD | DWELL | | | | | |
| | | 15 | 40 | | | | | |

- Tapping (TT, T two-spindle type) <Ver. 3>

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Tapping | PROC(01) | | HEAD | STAT-PZ | END-PTZ | NOMINL-D | PITCH | |
| | 0 | | 3 | 79 | 38 | 39 | 43 | |
| | | TOOL-NO | CUT-SPD | DWELL | | | | T-CODE |
| | | 14 | 15 | 40 | | | | 18 |

- Tapping (T one-spindle type) <Ver. 2>

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Tapping | PROC(01) | | | END-PT | NOMINL-D | PITCH | TOOL-NO | T-CODE |
| | 0 | | | 38 | 39 | 43 | 14 | 18 |
| | | CUT-SPD | DWELL | | | | | |
| | | 15 | 40 | | | | | |

- Tapping (T one-spindle type) <Ver. 3>

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Tapping | PROC(01) | | | STAT-PZ | END-PTZ | NOMINL-D | PITCH | |
| | 0 | | | 79 | 38 | 39 | 43 | |
| | | TOOL-NO | CUT-SPD | DWELL | | | | T-CODE |
| | | 14 | 15 | 40 | | | | 18 |

- Single action (TT, T two-spindle type)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Single action | PROC(01) | | HEAD | TOOL-NO | CUT-SPD | FEEDRATE | | T-CODE |
| | 0 | | 3 | 14 | 15 | 7 | | 18 |

* For the T two-spindle type, UNIT is replaced with SPINDLE.

- Single action (T one-spindle type)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Single action | PROC(01) | | | TOOL-NO | CUT-SPD | FEEDRATE | | T-CODE |
| | *0* | | | *14* | *15* | *7* | | *18* |

- Single action II (TT, T two-spindle type) <Ver. 3>
  [With animated simulation]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Single action | PROC(01) | | HEAD | TOOL-NO | | | TYPE | T-CODE |
| | *0* | | *3* | *14* | | | *81* | *18* |

[Without animated simulation]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Single action | PROC(01) | | HEAD | TOOL-NO | | | | T-CODE |
| | *0* | | *3* | *14* | | | | *18* |

- Single action II (T one-spindle type) <Ver. 3>
  [With animated simulation]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Single action | PROC(01) | | | TOOL-NO | | | TYPE | T-CODE |
| | *0* | | | *14* | | | *81* | *18* |

[Without animated simulation]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Single action | PROC(01) | | | TOOL-NO | | | | T-CODE |
| | *0* | | | *14* | | | | *18* |

- Subprogram call (TT, T two-spindle type)
  [Use of custom macro disabled]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Subprogram call | PROC(01) | | HEAD | PROGRAM | | | | |
| | *0* | | *3* | *8* | | | | |

[Use of custom macro enabled]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Subprogram call | PROC(01) | | HEAD | PROGRAM | DATA1 | DATA2 | DATA3 | DATA4 |
| | *0* | | *3* | *8* | *44* | *45* | *46* | *47* |

- Subprogram call (T one-spindle type)
  [Use of custom macro disabled]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Subprogram call | PROC(01) | | | PROGRAM | | | | |
| | *0* | | | *8* | | | | |

[Use of custom macro enabled]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Subprogram call | PROC(01) | | | PROGRAM | DATA1 | DATA2 | DATA3 | DATA4 |
| | *0* | | | *8* | *44* | *45* | *46* | *47* |

- Subprogram call II (TT, T two-spindle type)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Subprogram call | PROC(01) | | HEAD | PROGRAM | DATA A | DATA B | DATA C | DATA I |
| | *0* | | *3* | *8* | *82* | *83* | *84* | *85* |
| | | DATA J | DATA K | DATA D | DATA E | DATA F | DATA H | DATA M |
| | | *86* | *87* | *88* | *89* | *90* | *91* | *92* |

- Subprogram call II (T one-spindle type)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Subprogram call | PROC(01) | | | PROGRAM | DATA A | DATA B | DATA C | DATA I |
| | *0* | | | *8* | *82* | *83* | *84* | *85* |
| | | DATA J | DATA K | DATA D | DATA E | DATA F | DATA H | DATA M |
| | | *86* | *87* | *88* | *89* | *90* | *91* | *92* |

- C-axis center drilling (TT, T two-spindle type) <Ver. 2>

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C-axis center drilling | PROC(01) | AREA | HEAD | NOMINL-D | TOOL-NO | REV/MIN | FEED/MIN | T-CODE |
| | *0* | *2* | *3* | *39* | *14* | *15* | *16* | *18* |
| | | DWELL | INTRVAL | | | | | |
| | | *40* | *9* | | | | | |

* For the T two-spindle type, UNIT is replaced with SPINDLE.

- C-axis center drilling (TT, T two-spindle type) <Ver. 3>

  [Center drilling, starting]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C-axis center drilling | PROC(01) | AREA | HEAD | MACHN-2 | HOLE-DIA | TOOL-NO | REV/MIN | T-CODE |
| | *0* | *2* | *3* | *1* | *39* | *14* | *15* | *18* |
| | | FEED/MIN | DWELL | INTERVAL | | | | |
| | | *16* | *40* | *9* | | | | |

  [Center drilling + chamfering, starting + chamfering]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C-axis center drilling | PROC(01) | AREA | HEAD | MACHN-2 | CHAMF-DIA | TOOL-NO | REV/MIN | T-CODE |
| | *0* | *2* | *3* | *1* | *80* | *14* | *15* | *18* |
| | | FEED/MIN | DWELL | INTERVAL | | | | |
| | | *16* | *40* | *9* | | | | |

- C-axis center drilling (T one-spindle type) <Ver. 2>

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C-axis center drilling | PROC(01) | AREA | | NOMINL-D | TOOL-NO | REV/MIN | FEED/MIN | T-CODE |
| | *0* | *2* | | *39* | *14* | *15* | *16* | *18* |
| | | DWELL | INTRVAL | | | | | |
| | | *40* | *9* | | | | | |

- C-axis center drilling (T one-spindle type) <Ver. 3>

  [Center drilling, starting]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C-axis center drilling | PROC(01) | AREA | | MACHN-2 | HOLE-DIA | TOOL-NO | REV/MIN | T-CODE |
| | *0* | *2* | | *1* | *39* | *14* | *15* | *18* |
| | | FEED/MIN | DWELL | INTERVAL | | | | |
| | | *16* | *40* | *9* | | | | |

  [Center drilling + chamfering, starting + chamfering]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C-axis center drilling | PROC(01) | AREA | | MACHN-2 | CHAMF-DIA | TOOL-NO | REV/MIN | T-CODE |
| | *0* | *2* | | *1* | *80* | *14* | *15* | *18* |
| | | FEED/MIN | DWELL | INTERVAL | | | | |
| | | *16* | *40* | *9* | | | | |

- C–axis Drilling (TT, T two-spindle type) <Ver.2>

  [Up to 40 tools registered]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C–axis Drilling | PROC(01) | AREA | HEAD | | HOLE-DIA | TOOL-NO | REV/MIN | T-CODE |
| | *0* | *2* | *3* | | *41* | *14* | *15* | *18* |
| | | FEED/MIN | DWELL | PATTERN | CUT-DPTH | INTERVAL | | |
| | | *16* | *40* | *4* | *30* | *0* | | |

  [Up to 99 tools registered: drilling]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C–axis Drilling | PROC(01) | AREA | HEAD | MACHIN-2 | HOLE-DIA | TOOL-NO | REV/MIN | T-CODE |
| | *0* | *2* | *3* | *1* | *41* | *14* | *15* | *18* |
| | | FEED/MIN | DWELL | PATTERN | CUT-DPTH | INTERVAL | | |
| | | *16* | *40* | *4* | *30* | *9* | | |

  [Up to 99 tools registered: reaming]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C–axis Drilling | PROC(01) | AREA | HEAD | MACHIN-2 | HOLE-DIA | TOOL-NO | REV/MIN | T-CODE |
| | *0* | *2* | *3* | *1* | *41* | *14* | *15* | *18* |
| | | FEED/MIN | DWELL | INTERVAL | | | | |
| | | *16* | *40* | *9* | | | | |

* For the T two-spindle type, UNIT is replaced with SPINDLE.

- C–axis Drilling (TT, T two-spindle type) <Ver.3>

[Drilling]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C–axis Drilling | PROC(01) | AREA | HEAD | MACHIN-2 | PATTERN | PROC-DIA | TOOL-NO | T-CODE |
| | *0* | *2* | *3* | *1* | *4* | *41* | *34* | *18* |
| | | REV/MIN | FEED/MIN | DWELL | CUT-DPTH | INTERVAL | | |
| | | *15* | *16* | *40* | *30* | *9* | | |

[Reaming]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C–axis Drilling | PROC(01) | AREA | HEAD | MACHIN-2 | PATTERN | PROC-DIA | CHAMFLNG | |
| | *0* | *2* | *3* | *1* | *4* | *41* | *78* | |
| | | TOOL-NO | REV/MIN | FEED/MIN | DWELL | INTERVAL | | T-CODE |
| | | *14* | *15* | *16* | *40* | *9* | | *18* |

[End milling]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C–axis Drilling | PROC(01) | AREA | HEAD | MACHIN-2 | PATTERN | PROC-DIA | TOOL-NO | T-CODE |
| | *0* | *2* | *3* | *1* | *4* | *41* | *14* | *18* |
| | | REV/MIN | FEED/MIN | DWELL | INTERVAL | | | |
| | | *15* | *16* | *40* | *9* | | | |

[Boring]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C–axis Drilling | PROC(01) | AREA | HEAD | MACHIN-2 | PROC-DIA | TOOL-NO | REV/MIN | T-CODE |
| | *0* | *2* | *3* | *1* | *4* | *14* | *15* | *18* |
| | | FEED/MIN | DWELL | SHIFT | INTERVAL | | | |
| | | *16* | *40* | *42* | *9* | | | |

- Drilling (T one-spindle type) <Ver.2>

[Up to 40 tools registered]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C–axis Drilling | PROC(01) | AREA | | | HOLE-DIA | TOOL-NO | REV/MIN | T-CODE |
| | *0* | *2* | | | *41* | *14* | *15* | *18* |
| | | FEED/MIN | DWELL | PATTERN | CUT-DPTH | INTERVAL | | |
| | | *16* | *40* | *4* | *30* | *9* | | |

[Up to 99 tools registered: drilling]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C–axis Drilling | PROC(01) | AREA | | MACHIN-2 | HOLE-DIA | TOOL-NO | REV/MIN | T-CODE |
| | *0* | *2* | | *1* | *41* | *14* | *15* | *18* |
| | | FEED/MIN | DWELL | PATTERN | CUT-DPTH | INTERVAL | | |
| | | *16* | *40* | *4* | *30* | *9* | | |

[Up to 99 tools registered: reaming]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C–axis Drilling | PROC(01) | AREA | | MACHIN-2 | HOLE-DIA | TOOL-NO | REV/MIN | T-CODE |
| | *0* | *2* | | *1* | *41* | *14* | *15* | *18* |
| | | FEED/MIN | DWELL | INTERVAL | | | | |
| | | *16* | *40* | *9* | | | | |

* For the T two-spindle type, UNIT is replaced with SPINDLE.

- Drilling (T one-spindle type) <Ver.3>

[Drilling]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C–axis Drilling | PROC(01) | AREA | | MACHIN-2 | PATTERN | PROC-DIA | TOOL-NO | T-CODE |
| | 0 | 2 | | 1 | 4 | 41 | 14 | 18 |
| | | REV/MIN | FEED/MIN | DWELL | CUT-DPTH | INTERVAL | | |
| | | 15 | 16 | 40 | 30 | 9 | | |

[Reaming]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C–axis Drilling | PROC(01) | AREA | | MACHIN-2 | PATTERN | PROC-DIA | CHAMFLNG | |
| | 0 | 2 | | 1 | 4 | 41 | 78 | |
| | | TOOL-NO | REV/MIN | FEED/MIN | DWELL | INTERVAL | | T-CODE |
| | | 14 | 15 | 16 | 40 | 9 | | 18 |

[End milling]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C–axis Drilling | PROC(01) | AREA | | MACHIN-2 | PATTERN | PROC-DIA | TOOL-NO | T-CODE |
| | 0 | 2 | | 1 | 4 | 41 | 14 | 18 |
| | | REV/MIN | FEED/MIN | DWELL | INTERVAL | | | |
| | | 15 | 16 | 40 | 9 | | | |

[Boring]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C–axis Drilling | PROC(01) | AREA | | MACHIN-2 | PROC-DIA | TOOL-NO | REV/MIN | T-CODE |
| | 0 | 2 | | 1 | 4 | 14 | 15 | 18 |
| | | FEED/MIN | DWELL | SHIFT | INTERVAL | | | |
| | | 16 | 40 | 42 | 9 | | | |

- C-axis tapping (TT, T two-spindle type)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C-axis tapping | PROC(01) | AREA | HEAD | NOMINL-D | PITCH | TOOL-D | REV/MIN | T-CODE |
| | 0 | 2 | 3 | 39 | 43 | 14 | 15 | 18 |
| | | FEED/MIN | DWELL | INTRVAL | | | | |
| | | 16 | 40 | 9 | | | | |

- C-axis tapping (T one-spindle type)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C-axis tapping | PROC(01) | AREA | | NOMINL-D | PITCH | TOOL-D | REV/MIN | T-CODE |
| | 0 | 2 | | 39 | 43 | 14 | 15 | 18 |
| | | FEED/MIN | DWELL | INTRVAL | | | | |
| | | 16 | 40 | 9 | | | | |

- C-axis grooving (TT, T two-spindle type)

[Spindle positioning type : Only standard grooving on the side face enabled]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C-axis grooving | PROC(01) | | HEAD | GRV-DIA | TOOL-NO | REV/MIN | FEEDRATE1 | T-CODE |
| (rough) | 0 | | 3 | 48 | 14 | 15 | 49 | 18 |
| | | FEEDRATE2 | BEVEL-AM | | | | | |
| | | 51 | 53 | | | | | |
| C-axis grooving | PROC(01) | | HEAD | GRV-DIA | TOOL-NO | REV/MIN | FEED/MIN | T-CODE |
| (chamfer) | 0 | | 3 | 48 | 14 | 15 | 16 | 18 |
| | | BEVEL-AM | | | | | | |
| | | 53 | | | | | | |

[Contour control type - machining area: End face]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C-axis grooving | PROC(01) | AREA | HEAD | GRV-DIA | TOOL-NO | REV/MIN | FEEDRATE1 | T-CODE |
| (rough) | 0 | 2 | 3 | 48 | 14 | 15 | 49 | 18 |
| | | FEEDRATE2 | SHAPE | BEVEL-AM | | | | |
| | | 51 | 10 | 53 | | | | |
| C-axis grooving | PROC(01) | AREA | HEAD | GRV-DIA | TOOL-NO | REV/MIN | FEED/MIN | T-CODE |
| (chamfer) | 0 | 2 | 3 | 48 | 14 | 15 | 16 | 18 |
| | | SHAPE | BEVEL-AM | | | | | |
| | | 10 | 53 | | | | | |

* For the T two-spindle type, UNIT is replaced with SPINDLE.

[Contour control type - machining area:  Side face]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C-axis grooving | PROC(01) | AREA | HEAD | GRV-DIA | TOOL-NO | REV/MIN | FEEDRATE1 | T-CODE |
| (rough) | 0 | 2 | 3 | 48 | 14 | 15 | 49 | 18 |
| | | FEEDRATE2 | SHAPE | BEVEL-AM | | | | |
| | | 51 | 10 | 53 | | | | |
| C-axis grooving | PROC(01) | AREA | HEAD | GRV-DIA | TOOL-NO | REV/MIN | FEED/MIN | T-CODE |
| (chamfer) | 0 | 2 | 3 | 48 | 14 | 15 | 16 | 18 |
| | | SHAPE | BEVEL-AM | | | | | |
| | | 10 | 53 | | | | | |

- C-axis grooving (T one-spindle type)
  [Spindle positioning type : Only standard grooving on the side face enabled]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C-axis grooving | PROC(01) | | | GRV-DIA | TOOL-NO | REV/MIN | FEEDRATE1 | T-CODE |
| (rough) | 0 | | | 48 | 14 | 15 | 49 | 18 |
| | | FEEDRATE2 | BEVEL-AM | | | | | |
| | | 51 | 53 | | | | | |
| C-axis grooving | PROC(01) | | | GRV-DIA | TOOL-NO | REV/MIN | FEED/MIN | T-CODE |
| (chamfer) | 0 | | | 48 | 14 | 15 | 16 | 18 |
| | | BEVEL-AM | | | | | | |
| | | 53 | | | | | | |

[Contour control type - machining area : End face]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C-axis grooving | PROC(01) | AREA | | GRV-DIA | TOOL-NO | REV/MIN | FEEDRATE1 | T-CODE |
| (rough) | 0 | 2 | | 48 | 14 | 15 | 49 | 18 |
| | | FEEDRATE2 | SHAPE | BEVEL-AM | | | | |
| | | 51 | 10 | 53 | | | | |
| C-axis grooving | PROC(01) | AREA | | GRV-DIA | TOOL-NO | REV/MIN | FEED/MIN | T-CODE |
| (chamfer) | 0 | 2 | | 48 | 14 | 15 | 16 | 18 |
| | | SHAPE | BEVEL-AM | | | | | |
| | | 10 | 53 | | | | | |

[Contour control type - machining area : Side face]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C-axis grooving | PROC(01) | AREA | | GRV-DIA | TOOL-NO | REV/MIN | FEEDRATE1 | T-CODE |
| (rough) | 0 | 2 | | 48 | 14 | 15 | 49 | 18 |
| | | FEEDRATE2 | SHAPE | BEVEL-AM | | | | |
| | | 51 | 10 | 53 | | | | |
| C-axis grooving | PROC(01) | AREA | | GRV-DIA | TOOL-NO | REV/MIN | FEED/MIN | T-CODE |
| (chamfer) | 0 | 2 | | 48 | 14 | 15 | 16 | 18 |
| | | SHAPE | BEVEL-AM | | | | | |
| | | 10 | 53 | | | | | |

- C-axis notching (TT, T two-spindle type)
  [Machining area:  End face]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C-axis notching | PROC(01) | AREA | HEAD | STAT-PZ | END-PTZ | REMOVALX | TOOL-NO | T-CODE |
| (rough) | 0 | 2 | 3 | 54 | 55 | 25 | 14 | 18 |
| | | REV/MIN | FEED/MIN | FINISHX | FINISHZ | | | |
| | | 15 | 16 | 56 | 57 | | | |
| C-axis notching | PROC(01) | AREA | HEAD | STAT-PZ | END-PTZ | TOOL-NO | REV/MIN | T-CODE |
| (finish) | 0 | 2 | 3 | 54 | 55 | 14 | 15 | 18 |
| | | FEED/MIN | BEVEL-AM | | | | | |
| | | 16 | 53 | | | | | |
| C-axis notching | PROC(01) | AREA | HEAD | STAT-PZ | TOOL-NO | REV/MIN | FEED/MIN | T-CODE |
| (chamfer) | 0 | 2 | 3 | 54 | 14 | 15 | 16 | 18 |
| | | BEVEL-AM | | | | | | |
| | | 53 | | | | | | |

* For the T two-spindle type, UNIT is replaced with SPINDLE.

[Machining area: Side face]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C-axis notching | PROC(01) | AREA | HEAD | STAT-PZ | END-PTZ | REMOVALX | TOOL-NO | T-CODE |
| (rough) | 0 | 2 | 3 | 54 | 55 | 25 | 14 | 18 |
| | | REV/MIN | FEED/MIN | FINISHX | FINISHZ | | | |
| | | 15 | 16 | 56 | 57 | | | |
| C-axis notching | PROC(01) | AREA | HEAD | STAT-PZ | END-PTZ | TOOL-NO | REV/MIN | T-CODE |
| (finish) | 0 | 2 | 3 | 54 | 55 | 14 | 15 | 18 |
| | | FEED/MIN | | | | | | |
| | | 16 | | | | | | |

- C-axis notching (T one-spindle type)

[Machining area: End face]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C-axis notching | PROC(01) | AREA | | STAT-PZ | END-PTZ | REMOVALX | TOOL-NO | T-CODE |
| (rough) | 0 | 2 | | 54 | 55 | 25 | 14 | 18 |
| | | REV/MIN | FEED/MIN | FINISHX | FINISHZ | | | |
| | | 15 | 16 | 56 | 57 | | | |
| C-axis notching | PROC(01) | AREA | | STAT-PZ | END-PTZ | TOOL-NO | REV/MIN | T-CODE |
| (finish) | 0 | 2 | | 54 | 55 | 14 | 15 | 18 |
| | | FEED/MIN | BEVEL-AM | | | | | |
| | | 16 | 53 | | | | | |
| C-axis notching | PROC(01) | AREA | | STAT-PZ | TOOL-NO | REV/MIN | FEED/MIN | T-CODE |
| (chamfer) | 0 | 2 | | 54 | 14 | 15 | 16 | 18 |
| | | BEVEL-AM | | | | | | |
| | | 53 | | | | | | |

[Machining area: Side face]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C-axis notching | PROC(01) | AREA | | STAT-PZ | END-PTZ | REMOVALX | TOOL-NO | T-CODE |
| (rough) | 0 | 2 | | 54 | 55 | 25 | 14 | 18 |
| | | REV/MIN | FEED/MIN | FINISHX | FINISHZ | | | |
| | | 15 | 16 | 56 | 57 | | | |
| C-axis notching | PROC(01) | AREA | | STAT-PZ | END-PTZ | TOOL-NO | REV/MIN | T-CODE |
| (finish) | 0 | 2 | | 54 | 55 | 14 | 15 | 18 |
| | | FEED/MIN | | | | | | |
| | | 16 | | | | | | |

- C-axis cylindrical machining (TT, T two-spindle type)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C-axis cylindrical | PROC(01) | | HEAD | STAT-PX | DEPTH | GRV-DIA | TOOL-NO | T-CODE |
| machining (rough) | 0 | | 3 | 58 | 59 | 48 | 14 | 18 |
| | | REV/MIN | FEEDRATE1 | FEEDRATE2 | BEVEL-AM | | | |
| | | 15 | 50 | 52 | 53 | | | |
| C-axis cylindrical | PROC(01) | | HEAD | STAT-PX | DEPTH | GRV-DIA | TOOL-NO | T-CODE |
| machining | 0 | | 3 | 58 | 59 | 48 | 14 | 18 |
| (chamfer) | | REV/MIN | FEED/MIN | BEVEL-AM | | | | |
| | | 15 | 16 | 53 | | | | |

- C-axis cylindrical machining (T one-spindle type)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C-axis cylindrical | PROC(01) | | | STAT-PX | DEPTH | GRV-DIA | TOOL-NO | T-CODE |
| machining (rough) | 0 | | | 58 | 59 | 48 | 14 | 18 |
| | | REV/MIN | FEEDRATE1 | FEEDRATE2 | BEVEL-AM | | | |
| | | 15 | 50 | 52 | 53 | | | |
| C-axis cylindrical | PROC(01) | | | STAT-PX | DEPTH | GRV-DIA | TOOL-NO | T-CODE |
| machining | 0 | | | 58 | 59 | 48 | 14 | 18 |
| (chamfer) | | REV/MIN | FEED/MIN | BEVEL-AM | | | | |
| | | 15 | 16 | 53 | | | | |

* For the T two-spindle type, UNIT is replaced with SPINDLE.

- Transfer

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Transfer | PROC(01) | User Program | | | | | | | |
| | *0* | | | | | | | | |

- Auxiliary

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Auxiliary | PROC(01) | User Program | | | | | | | |
| | *0* | | | | | | | | |

- M (TT, T two-spindle type)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| M | PROC(01) | | HEAD | M | M | M | M | M | |
| | *0* | | *3* | *60* | *60* | *60* | *60* | *60* | |

- M (T one-spindle type)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| M | PROC(01) | | | M | M | M | M | M | |
| | *0* | | | *60* | *60* | *60* | *60* | *60* | |

- End

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| End | PROC(01) | RETURN | | RETURN-X | RETURN-Z | RETURN-C | END-M | LOOP |
| | *0* | *11* | | *61* | *62* | *63* | *12* | *64* |

- Y-axis center drilling (TT, T two-spindle type)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Y-axis center drilling | PROC(01) | AREA | HEAD | NOMINL-D | TOOL-NO | REV/MIN | FEED/MIN | T-CODE |
| | *0* | *2* | *3* | *39* | *14* | *15* | *16* | *18* |
| | | DWELL | HOLE-PTN | | | | | |
| | | *40* | *69* | | | | | |

- Y-axis center drilling (T one-spindle type)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Y-axis center drilling | PROC(01) | AREA | | NOMINL-D | TOOL-NO | REV/MIN | FEED/MIN | T-CODE |
| | *0* | *2* | | *39* | *14* | *15* | *16* | *18* |
| | | DWELL | HOLE-PTN | | | | | |
| | | *40* | *69* | | | | | |

- Y-axis drilling (TT, T two-spindle type)
  [Up to 40 tools registered]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Y-axis drilling | PROC(01) | AREA | HEAD | | HOLE-DIA | TOOL-NO | REV/MIN | T-CODE |
| | *0* | *2* | *3* | | *41* | *14* | *15* | *18* |
| | | FEED/MIN | DWELL | PATTERN | CUT-DPTH | HOLE-PTN | | |
| | | *16* | *40* | *4* | *30* | *69* | | |

[Up to 99 tools registered: drilling]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Y-axis drilling | PROC(01) | AREA | HEAD | MACHN-2 | HOLE-DIA | TOOL-NO | REV/MIN | T-CODE |
| | *0* | *2* | *3* | *1* | *41* | *14* | *15* | *18* |
| | | FEED/MIN | DWELL | PATTERN | CUT-DPTH | HOLE-PTN | | |
| | | *16* | *40* | *4* | *30* | *69* | | |

[Up to 99 tools registered: reaming]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Y-axis drilling | PROC(01) | AREA | HEAD | MACHN-2 | HOLE-DIA | TOOL-NO | REV/MIN | T-CODE |
| | *0* | *2* | *3* | | *41* | *14* | *15* | *18* |
| | | FEED/MIN | DWELL | HOLE-PTN | | | | |
| | | *16* | *40* | *69* | | | | |

- Y-axis drilling (T one-spindle type)
  [Up to 40 tools registered]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Y-axis drilling | PROC(01) | AREA | | | HOLE-DIA | TOOL-NO | REV/MIN | T-CODE |
| | *0* | *2* | | | *41* | *14* | *15* | *18* |
| | | FEED/MIN | DWELL | PATTERN | CUT-DPTH | HOLE-PTN | | |
| | | *16* | *40* | *4* | *30* | *69* | | |

* For the T two-spindle type, UNIT is replaced with SPINDLE.

[Up to 99 tools registered:  drilling]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Y-axis drilling | PROC(01) | AREA | | MACHN-2 | HOLE-DIA | TOOL-NO | REV/MIN | T-CODE |
| | *0* | *2* | | *1* | *41* | *14* | *15* | *18* |
| | | FEED/MIN | DWELL | PATTERN | CUT-DPTH | HOLE-PTN | | |
| | | *16* | *40* | *4* | *30* | *69* | | |

[Up to 99 tools registered:  reaming]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Y-axis drilling | PROC(01) | AREA | | MACHN-2 | HOLE-DIA | TOOL-NO | REV/MIN | T-CODE |
| | *0* | *2* | | *1* | *41* | *14* | *15* | *18* |
| | | FEED/MIN | DWELL | HOLE-PTN | | | | |
| | | *16* | *40* | *69* | | | | |

- Y-axis tapping (TT, T two-spindle type)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Y-axis tapping | PROC(01) | AREA | HEAD | NOMINL-D | PITCH | TOOL-NO | REV/MIN | T-CODE |
| | *0* | *2* | *3* | *39* | *43* | *14* | *15* | *18* |
| | | FEED/MIN | DWELL | HOLE-PTN | | | | |
| | | *16* | *40* | *69* | | | | |

- Y-axis tapping (T one-spindle type)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Y-axis tapping | PROC(01) | AREA | | NOMINL-D | PITCH | TOOL-NO | REV/MIN | T-CODE |
| | *0* | *2* | | *39* | *43* | *14* | *15* | *18* |
| | | FEED/MIN | DWELL | HOLE-PTN | | | | |
| | | *16* | *40* | *69* | | | | |

- Y milling (TT, T two-spindle type)
  [Machining area:  End face]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Y milling | PROC(01) | AREA | HEAD | DEPTH | GRV-DIA | TOOL-NO | REV/MIN | T-CODE |
| | *0* | *2* | *3* | *59* | *48* | *14* | *15* | *18* |
| | | FEEDRATE1 | FEEDRATE2 | ESCAPE-SZ | SHIFT-DR | | | |
| | | *73* | *51* | *75* | *70* | | | |

[Machining area:  Side face]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Ymilling | PROC(01) | AREA | HEAD | DEPTH | GRV-DIA | TOOL-NO | REV/MIN | T-CODE |
| | *0* | *2* | *3* | *59* | *48* | *14* | *15* | *18* |
| | | FEEDRATE1 | FEEDRATE2 | ESCAPE-SX | SHIFT-DR | | | |
| | | *74* | *51* | *76* | *70* | | | |

- Y milling (T one-spindle type)
  [Machining area:  End face]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Y milling | PROC(01) | AREA | | DEPTH | GRV-DIA | TOOL-NO | REV/MIN | T-CODE |
| | *0* | *2* | | *59* | *48* | *14* | *15* | *18* |
| | | FEEDRATE1 | FEEDRATE2 | ESCAPE-SZ | SHIFT-DR | | | |
| | | *73* | *51* | *75* | *70* | | | |

[Machining area:  Side face]

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Ymilling | PROC(01) | AREA | | DEPTH | GRV-DIA | TOOL-NO | REV/MIN | T-CODE |
| | *0* | *2* | | *59* | *48* | *14* | *15* | *18* |
| | | FEEDRATE1 | FEEDRATE2 | ESCAPE-SX | SHIFT-DR | | | |
| | | *74* | *51* | *76* | *70* | | | |

* For the T two-spindle type, UNIT is replaced with SPINDLE.

- Figure data

<Residual machining>

| | | | | | |
|---|---|---|---|---|---|
| STAT-PX | STAT-PZ | END-PTX | END-PTZ | ROUND | CHAMFER |
| *10* | *11* | *12* | *13* | *14* | *15* |
| STAT-PX | STAT-PZ | END-PTX | END-PTZ | ROUND | CHAMFER |
| *20* | *21* | *22* | *23* | *24* | *25* |
| STAT-PX | STAT-PZ | END-PTX | END-PTZ | ROUND | CHAMFER |
| *30* | *31* | *32* | *33* | *34* | *35* |
| STAT-PX | STAT-PZ | END-PTX | END-PTZ | ROUND | CHAMFER |
| *40* | *41* | *42* | *43* | *44* | *45* |
| STAT-PX | STAT-PZ | END-PTX | END-PTZ | ROUND | CHAMFER |
| *50* | *51* | *52* | *53* | *54* | *55* |

<Threading-general-purpose>

| | | | |
|---|---|---|---|
| STAT-PX | STAT-PZ | END-PTX | END-PTZ |
| *10* | *11* | *12* | *13* |
| STAT-PX | STAT-PZ | END-PTX | END-PTZ |
| *20* | *21* | *22* | *23* |
| STAT-PX | STAT-PZ | END-PTX | END-PTZ |
| *30* | *31* | *32* | *33* |
| STAT-PX | STAT-PZ | END-PTX | END-PTZ |
| *40* | *41* | *42* | *43* |
| STAT-PX | STAT-PZ | END-PTX | END-PTZ |
| *50* | *51* | *52* | *53* |

<Threading-metric>

| | | |
|---|---|---|
| THRD DIA | STAT-PZ | END-PTZ |
| *14* | *11* | *13* |
| THRD DIA | STAT-PZ | END-PTZ |
| *24* | *21* | *23* |
| THRD DIA | STAT-PZ | END-PTZ |
| *34* | *31* | *33* |
| THRD DIA | STAT-PZ | END-PTZ |
| *44* | *41* | *43* |
| THRD DIA | STAT-PZ | END-PTZ |
| *54* | *51* | *53* |

<Threading-PT>

| | | |
|---|---|---|
| END-PTX | STAT-PZ | END-PTZ |
| *10* | *11* | *12* |

<Threading-unified, PF>

| | | |
|---|---|---|
| THRD DIA | STAT-PZ | END-PTZ |
| *14* | *11* | *12* |

<Grooving-standard>

| | | | | | | |
|---|---|---|---|---|---|---|
| STAT-PX | STAT-PZ | END-PTX | WIDTH | DEPTH | PITCH | NUMBER |
| *0* | *1* | *2* | *4* | *5* | *6* | *7* |
| CHAMFER | | | | | | |
| *8* | | | | | | |

<Grooving-slanted>

| | | | | | |
|---|---|---|---|---|---|
| STAT-PX | STAT-PZ | WIDTH | GRV-DIA | PITCH | NUMBER |
| *0* | *1* | *4* | *5* | *6* | *7* |

<Grooving-trapezoidal>

| | | | |
|---|---|---|---|
| PITCH | NUMBER | | |
| *6* | *7* | | |
| STAT-PX | STAT-PZ | | |
| *0* | *1* | | |
| POINT 1X | POINT 1Z | ROUND | CHAMFER |
| *10* | *11* | *12* | *13* |
| POINT 2X | POINT 2Z | ROUND | CHAMFER |
| *20* | *21* | *22* | *23* |
| POINT 3X | POINT 3Z | ROUND | CHAMFER |
| *30* | *31* | *32* | *33* |
| POINT 4X | POINT 4Z | ROUND | CHAMFER |
| *40* | *41* | *42* | *43* |
| END-PTX | END-PTZ | | |
| *2* | *3* | | |

<Grooving-thread>

| STAT-PX | STAT-PZ | END-PTX | WIDTH | GRV-DIA | CHAMFER |
|---|---|---|---|---|---|
| *0* | *1* | *2* | *4* | *5* | *8* |

<Grooving-thread> (when a subspindle of T one-spindle type is selected)

| STAT-PX | STAT-PZ | END-PTX | WIDTH | GRV-DIA | CHAMFER |
|---|---|---|---|---|---|
| *0* | *1* | *2* | *4* | *5* | *8* |

<Necking>

| | | | | | |
|---|---|---|---|---|---|
| BSC PTX1 | BSC PTZ1 | BSC PTX2 | BSC PTZ2 | BSC PTX3 | BSC PTZ3 |
| *0* | *1* | *2* | *3* | *4* | *5* |
| BSC PTX4 | BSC PTZ4 | BSC PTX5 | BSC PTZ5 | | |
| *6* | *7* | *8* | *9* | | |

<C-axis center drilling, C-axis drilling, C-axis tapping-[constant interval]>

| STAT-PX | STAT-PZ | STAT-PC | DEPTH | ANGLE | ITEMS | FIN-ANGL |
|---|---|---|---|---|---|---|
| *0* | *1* | *2* | *3* | *4* | *5* | *6* |

<C-axis center drilling, C-axis drilling, C-axis tapping-[variable interval-end face]>

| | | | |
|---|---|---|---|
| POINT 1X | POINT 1C | STAT-PZ | DEPTH |
| *10* | *11* | *12* | *13* |
| POINT 2X | POINT 2C | STAT-PZ | DEPTH |
| *20* | *21* | *22* | *23* |
| POINT 3X | POINT 3C | STAT-PZ | DEPTH |
| *30* | *31* | *32* | *33* |
| POINT 4X | POINT 4C | STAT-PZ | DEPTH |
| *40* | *41* | *42* | *43* |
| POINT 5X | POINT 5C | STAT-PZ | DEPTH |
| *50* | *51* | *52* | *53* |
| POINT 6X | POINT 6C | STAT-PZ | DEPTH |
| *60* | *61* | *62* | *63* |

<C-axis center drilling, C-axis drilling, C-axis tapping-[variable interval-side face]>

| | | | |
|---|---|---|---|
| POINT 1Z | POINT 1C | STAT-PX | DEPTH |
| *10* | *11* | *12* | *13* |
| POINT 2Z | POINT 2C | STAT-PX | DEPTH |
| *20* | *21* | *22* | *23* |
| POINT 3Z | POINT 3C | STAT-PX | DEPTH |
| *30* | *31* | *32* | *33* |
| POINT 4Z | POINT 4C | STAT-PX | DEPTH |
| *40* | *41* | *42* | *43* |
| POINT 5Z | POINT 5C | STAT-PX | DEPTH |
| *50* | *51* | *52* | *53* |
| POINT 6Z | POINT 6C | STAT-PX | DEPTH |
| *60* | *61* | *62* | *63* |

<C-axis grooving-[constant interval]>

| | | | | | | |
|---|---|---|---|---|---|---|
| STAT-PX | STAT-PZ | STAT-PC | DEPTH | GRV-LENG | ANGLE | ITEMS |
| *0* | *1* | *2* | *3* | *4* | *5* | *6* |
| FIN-ANGL | | | | | | |
| *7* | | | | | | |

<C-axis grooving-[variable interval-end face]>

| | | | | | |
|---|---|---|---|---|---|
| STAT-PX | STAT-PZ | STAT-PC | END-PTX | END-PTC | DEPTH |
| *10* | *11* | *12* | *13* | *14* | *15* |
| STAT-PX | STAT-PZ | STAT-PC | END-PTX | END-PTC | DEPTH |
| *20* | *21* | *22* | *23* | *24* | *25* |
| STAT-PX | STAT-PZ | STAT-PC | END-PTX | END-PTC | DEPTH |
| *30* | *31* | *32* | *33* | *34* | *35* |
| STAT-PX | STAT-PZ | STAT-PC | END-PTX | END-PTC | DEPTH |
| *40* | *41* | *42* | *43* | *44* | *45* |
| STAT-PX | STAT-PZ | STAT-PC | END-PTX | END-PTC | DEPTH |
| *50* | *51* | *52* | *53* | *54* | *55* |
| STAT-PX | STAT-PZ | STAT-PC | END-PTX | END-PTC | DEPTH |
| *60* | *61* | *62* | *63* | *64* | *65* |

<C-axis grooving-[variable interval-side face]>

| | | | | | |
|---|---|---|---|---|---|
| STAT-PX | STAT-PZ | STAT-PC | END-PTZ | END-PTC | DEPTH |
| *10* | *11* | *12* | *13* | *14* | *15* |
| STAT-PX | STAT-PZ | STAT-PC | END-PTZ | END-PTC | DEPTH |
| *20* | *21* | *22* | *23* | *24* | *25* |
| STAT-PX | STAT-PZ | STAT-PC | END-PTZ | END-PTC | DEPTH |
| *30* | *31* | *32* | *33* | *34* | *35* |
| STAT-PX | STAT-PZ | STAT-PC | END-PTZ | END-PTC | DEPTH |
| *40* | *41* | *42* | *43* | *44* | *45* |
| STAT-PX | STAT-PZ | STAT-PC | END-PTZ | END-PTC | DEPTH |
| 50 | 51 | 52 | 53 | 54 | 55 |
| STAT-PX | STAT-PZ | STAT-PC | END-PTZ | END-PTC | DEPTH |
| *60* | *61* | *62* | *63* | *64* | *65* |

<Y-axis center drilling, Y-axis drilling, Y-axis tapping-[circle-end face]>

| | | | | | |
|---|---|---|---|---|---|
| CENTR-PX | CENTR-PY | STAT-PZ | STAT-ANGL | DEPTH | REVOLVE-A |
| *0* | *1* | *2* | *3* | *4* | *5* |
| RADIUS | ANGLE | ITEMS | FIN-ANGL | | |
| *6* | *7* | *8* | *9* | | |

<Y-axis center drilling, Y-axis drilling, Y-axis tapping-[circle-side face]>

| | | | | | |
|---|---|---|---|---|---|
| CENTR-PY | CENTR-PZ | STAT-PX | STAT-ANGL | DEPTH | REVOLVE-A |
| *0* | *1* | *2* | *3* | *4* | *5* |
| RADIUS | ANGLE | ITEMS | FIN-ANGL | | |
| *6* | *7* | *8* | *9* | | |

<Y-axis center drilling, Y-axis drilling, Y-axis tapping-[grid-end face]>

| | | | | | |
|---|---|---|---|---|---|
| CENTR-PX | CENTR-PY | STAT-PZ | DEPTH | REVOLVE-A | PITCH-WID |
| *0* | *1* | *2* | *3* | *4* | *5* |
| COUNT/LIN | REMOV-WID | LIN-COUNT | | | |
| *6* | *7* | *8* | | | |

<Y-axis center drilling, Y-axis drilling, Y-axis tapping-[grid-side face]>

| | | | | | |
|---|---|---|---|---|---|
| CENTR-PY | CENTR-PZ | STAT-PX | DEPTH | REVOLVE-A | PITCH-WID |
| *0* | *1* | *2* | *3* | *4* | *5* |
| COUNT/LIN | REMOV-WID | LIN-COUNT | | | |
| *6* | *7* | *8* | | | |

<Y-axis center drilling, Y-axis drilling, Y-axis tapping-[desired-end face]>

| | | | | |
|---|---|---|---|---|
| POINT 1X | POINT 1Y | POINT 1C | POINT 1Z | DEPTH |
| *10* | *11* | *12* | *13* | *14* |
| POINT 2X | POINT 2Y | POINT 2C | POINT 2Z | DEPTH |
| *20* | *21* | *22* | *23* | *24* |
| POINT 3X | POINT 3Y | POINT 3C | POINT 3Z | DEPTH |
| *30* | *31* | *32* | *33* | *34* |
| POINT 4X | POINT 4Y | POINT 4C | POINT 4Z | DEPTH |
| *40* | *41* | *42* | *43* | *44* |
| POINT 5X | POINT 5Y | POINT 5C | POINT 5Z | DEPTH |
| *50* | *51* | *52* | *53* | *54* |
| POINT 6X | POINT 6Y | POINT 6C | POINT 6Z | DEPTH |
| *60* | *61* | *62* | *63* | *64* |

<Y-axis center drilling, Y-axis drilling, Y-axis tapping-[desired-side face]>

| | | | | |
|---|---|---|---|---|
| POINT 1Y | POINT 1Z | POINT 1C | POINT 1X | DEPTH |
| *10* | *11* | *12* | *13* | *14* |
| POINT 2Y | POINT 2Z | POINT 2C | POINT 2X | DEPTH |
| *20* | *21* | *22* | *23* | *24* |
| POINT 3Y | POINT 3Z | POINT 3C | POINT 3X | DEPTH |
| *30* | *31* | *32* | *33* | *34* |
| POINT 4Y | POINT 4Z | POINT 4C | POINT 4X | DEPTH |
| *40* | *41* | *42* | *43* | *44* |
| POINT 5Y | POINT 5Z | POINT 5C | POINT 5X | DEPTH |
| *50* | *51* | *52* | *53* | *54* |
| POINT 6Y | POINT 6Z | POINT 6C | POINT 6X | DEPTH |
| *60* | *61* | *62* | *63* | *64* |

- Contour data

<Bar machining, pattern repeating>

| | | | | | |
|---|---|---|---|---|---|
| 0 | START | X= | Z= | | |
| 0 | ← | X= | Z= | SR= | |
| 0 | → | X= | Z= | SR= | |
| 0 | ↑ | X= | Z= | SR= | |
| 0 | ↓ | X= | Z= | SR= | |
| 0 | ↗ | X= | Z= | SR= | |
| 0 | ↖ | X= | Z= | SR= | |
| 0 | ↙ | X= | Z= | SR= | |
| 0 | ↘ | X= | Z= | SR= | |
| 0 | ↻ | X= | Z= | R= | SR= |
| 0 | ↺ | X= | Z= | R= | SR= |
| 0 | CHAMFR | X= | Z= | SR= | |
| 0 | ROUND | X= | Z= | R= | SR= |

<C-axis notching>

| | | | | |
|---|---|---|---|---|
| 0 | START | X= | Z= | |
| 0 | APROCH | X= | Z= | R= |
| 0 | ← | X= | Z= | |
| 0 | → | X= | Z= | |
| 0 | ↑ | X= | Z= | |
| 0 | ↓ | X= | Z= | |
| 0 | ↗ | X= | Z= | |
| 0 | ↖ | X= | Z= | |
| 0 | ↙ | X= | Z= | |
| 0 | ↘ | X= | Z= | |
| 0 | ↻ | X= | Z= | R= |
| 0 | ↺ | X= | Z= | R= |
| 0 | CHAMFR | X= | Z= | |
| 0 | ROUND | X= | Z= | R= |
| 0 | ESCAPE | X= | Z= | R= |

<C-axis cylindrical machining>

| | | | | |
|---|---|---|---|---|
| 0 | START | C= | Z= | |
| 0 | ← | C= | Z= | |
| 0 | → | C= | Z= | |
| 0 | ↑ | C= | Z= | |
| 0 | ↓ | C= | Z= | |
| 0 | ↗ | C= | Z= | |
| 0 | ↖ | C= | Z= | |
| 0 | ↙ | C= | Z= | |
| 0 | ↘ | C= | Z= | |
| 0 | ↻ | C= | Z= | R= |
| 0 | ↺ | C= | Z= | R= |
| 0 | CHAMFR | C= | Z= | |
| 0 | ROUND | C= | Z= | R= |
| 0 | SHIFT | C= | Z= | R= |

<Y-axis milling-end face>

| | | | | | |
|---|---|---|---|---|---|
| *0* | START | X= | Y= | C= | |
| *0* | ← | X= | Y= | | |
| *0* | → | X= | Y= | | |
| *0* | ↑ | X= | Y= | | |
| *0* | ↓ | X= | Y= | | |
| *0* | ↗ | X= | Y= | | |
| *0* | ↖ | X= | Y= | | |
| *0* | ↙ | X= | Y= | | |
| *0* | ↘ | X= | Y= | | |
| *0* | ↻ | X= | Z= | R= | |
| *0* | ↺ | X= | Z= | R= | |
| *0* | CHAMFR | X= | Y= | | |
| *0* | ROUND | X= | Y= | R= | |
| *0* | SHIFT | X= | Y= | C= | Z= |

<Y-axis milling-side face>

| | | | | | |
|---|---|---|---|---|---|
| *0* | START | Y= | Z= | | |
| *0* | ← | Y= | Z= | | |
| *0* | → | Y= | Z= | | |
| *0* | ↑ | Y= | Z= | | |
| *0* | ↓ | Y= | Z= | | |
| *0* | ↗ | Y= | Z= | | |
| *0* | ↖ | Y= | Z= | | |
| *0* | ↙ | Y= | Z= | | |
| *0* | ↘ | Y= | Z= | | |
| *0* | ↻ | X= | Z= | R= | |
| *0* | ↺ | X= | Z= | R= | |
| *0* | CHAMFR | Y= | Z= | | |
| *0* | ROUND | Y= | Z= | R= | |
| *0* | SHIFT | Y= | Z= | C= | X= |

- Action figure data

<Single action>

| | | | | | |
|---|---|---|---|---|---|
| *0* | AUX-F | M=1 | S=2 | T=3 | |
| *0* | POSITN | X=1 | Z=2 | | |
| *0* | LINE | X=1 | Z=2 | F=3 | |
| *0* | ARC ↻ | X=1 | Z=2 | R=3 | F=4 |
| *0* | ARC ↺ | X=1 | Z=2 | R=3 | F=4 |
| *0* | DWELL | P=1 | | | |
| *0* | THREAD | X=1 | Z=2 | F=3 | |

<Single action II> <Ver. 3>

| | | | | |
|---|---|---|---|---|
| *0* | *1* | *2* | *3* | *4* |

(b) System variable indicating a data line : #9201

In system variable #9201, a value indicating the data line to which the cursor is positioned is set.

#9201
- 0 : Process data
- 1 : Figure data
- 2 : Contour data or single action figure data
- 3 : Initial setting data
- 4 : Workpiece point data (point on outer surface)
- 5 : Workpiece point data (point on inner surface)
- 6 : Program number or name

(c) System variable indicating contour data : #9202

In system variable #9202, a value indicating the line of contour data to which the cursor is positioned is set.

#9202
- 0 : START
  (Bar, Pattern repeating, C-axis notching, C-axis cylinder)
- 1 : →, ←
  (Bar, Pattern repeating, C-axis notching, C-axis cylinder)
- 2 : ↑, ↓
  (Bar, Pattern repeating, C-axis notching, C-axis cylinder)
- 3 : ↖, ↙, ↘, ↗
  (Bar, Pattern repeating, C-axis notching, C-axis cylinder)
- 4 : ↻, ↺
  (Bar, Pattern repeating, C-axis notching, C-axis cylinder)
- 5 : CHAMFER
  (Bar, Pattern repeating, C-axis notching, C-axis cylinder)
- 6 : ROUND
  (Bar, Pattern repeating, C-axis notching, C-axis cylinder)
- 7 : ESCAPE (C-axis nothcing)
- 8 : SHIFT (C-axis cylinder)
- 9 : APROACH (C-axis nothcing)
- 10: at inputting new figuredata in Single Action
- 11: AUX (Single action)
- 12: POSITION (Single action)
- 13: LINE (Single action)
- 14: ARC ↻ (Single action)
- 15: ARC ↺ (Single action)
- 16: DWELL (Single action)
- 17: THREAD (Single action)

(d) System variable indicating the number of lines of contour data : #9203

In system variable #9203, a value indicating the position of the line of the contour data or single action figure data, to which the cursor is positioned, is set. The line position is counted from the first line.

| | | | | | |
|---|---|---|---|---|---|
| *0* | START | X= | Z= | | |
| *1* | ↑ | X= | Z= | SR= | |
| *2* | ROUND | X= | Z= | R= | SR= |
| *3* | ← | X= | Z= | SR= | |
| *4* | ↖ | X= | Z= | SR= | |
| *5* | ← | X= | Z= | SR= | |
| *6* | CHAMFER | X= | Z= | SR= | |
| *7* | ↑ | X= | Z= | SR= | |
| | | ≀ | | | |
| *20* | ROUND | X= | Z= | R= | SR= |
| | | ≀ | | | |
| *30* | ↑ | X= | Z= | SR= | |

(e) System variables indicating the position of the cursor on the screen : #9108, #9109

In system variables #9108 and #9109, values indicating the position of the cursor on the screen are set as character coordinates.

#9108 : X coordinate of the cursor position
#9109 : Y coordinate of the cursor position

(3) System variable indicating a soft key page : #9204

In system variable #9204, a value indicating the currently displayed soft key page is set.

#9204 0 : The first page is displayed.
1 : The second page is displayed.
2 : The third page is displayed.
3 : The fourth page is displayed.

(4) System variable indicating the window page displayed by pressing the [+] soft key : #9205

The value of system variable #9205 is incremented each time the [+] soft key is pressed while the cursor is positioned to PROGRAM NO. or PROGRAM NAME. When the cursor is positioned for the first time, 0 is set. The user program can display a window according to this value.

If the window for the last page is displayed, the user program must set -1 in system variable #9205.

#9205 0 : The window for the first page is displayed.
1 : The window for the second page is displayed.
2 : The window for the third page is displayed.
⋮
10: The window for the eleventh page is displayed.
–1: This value must be set to notify the system that the last page is currently displayed.

(5) System variable to notify the system whether a window is being displayed by a user program : #9206

System variable #9206 is used to notify the system that a window is being displayed by a user program. The user can display the standard system window by setting an appropriate value in this variable.

#9206 0 : No window is displayed by a user program. Nor is the standard system window displayed, either.
1 : No window is displayed by a user program. The standard system window is displayed.
2 : A window is displayed by a user program.

## E.6.7 Using a User Program to Display Guidance on the Detailed Program Data Screen (Not Supported by Ver.1.)

### E.6.7.1 Overview

FANUC Super CAP T/CAP II T enable a user-created program to display the user's own guidance on part of the detailed program data screen.

### E.6.7.2 Details

(1) Number of the program used to display guidance on the detailed program data screen

O1023 is assigned to the program used to display guidance on the detailed program data screen. This program is called only once when a screen is changed to the detailed program data screen or when the cursor is moved.

(2) System variable indicating the position of the cursor

#9207 : Number of the item to which the cursor is positioned

| Classification | Item name | | Item No. | Classification | Item name | | Item No. |
|---|---|---|---|---|---|---|---|
| Common | PASSPOINT 1 | X1 | 1 | Drilling C-axis nothcing | DEPTH DECRS. | DC | 901 |
| | | Z1 | 2 | | RETURN AMNT. | RU | 902 |
| | PASSPOINT 2 | X2 | 3 | | MIN DEPTH | MI | 903 |
| | | Z2 | 4 | Subprogram call (Enabled only when bit 3 of parameter 9771 is set to 1) | DATA A | | 1301 |
| | START PNT. | CX | 5 | | DATA B | | 1302 |
| | (MACHINING START POINT) | CZ | 6 | | DATA C | | 1303 |
| | S-DRCT. | RS | 7 | | DATA I | | 1304 |
| | FINISHING | FX | 8 | | DATA J | | 1305 |
| | | FZ | 9 | | DATA K | | 1306 |
| | PROC MOVE. | MP | 10 | | DATA D | | 1307 |
| | SPINDLGEAR | GM | 11 | | DATA E | | 1308 |
| | COOLANT | CM | 12 | | DATA F | | 1309 |
| | MILLNGGEAR | GM | 13 | | DATA H | | 1310 |
| | PRG.OVERRIDE | OV | 14 | | DATA M | | 1311 |
| Bar amchining | RESIDUAL CUT | AU | 101 | | DATA Q | | 1312 |
| End facing | END POINT | EX | 401 | | DATA R | | 1313 |
| Threading | THREAD ANGLE | TA | 501 | | DATA S | | 1314 |
| | CUT METHOD (1) | P1 | 502 | | DATA T | | 1315 |
| | CUT METHOD (2) | P2 | 503 | | DATA U | | 1316 |
| | NO. OF THREAD | NT | 504 | | DATA V | | 1317 |
| | SPARK OUT | SO | 505 | | DATA W | | 1318 |
| | THRD HEIGHT | SH | 506 | | DATA X | | 1319 |
| Grooving | EXEC. CYCLE | EX | 601 | | DATA Y | | 1320 |
| Necking | WIDTH | WT | 701 | | DATA Z | | 1321 |
| | DEPTH | DT | 702 | C-axis nothcing | CUT WID (%) | Q | 1801 |
| | ROUND | R | 703 | Y-axis drilling | SKIP POINT 1 | | 2401 |
| | APPROACH ANG | AA | 704 | | SKIP POINT 2 | | 2402 |
| | RELIEF AMONT | W1 | 705 | | SKIP POINT 3 | | 2403 |
| | RELIEF ANGLE | WA | 706 | | | | |

(3) System variable indicating whether the program detail data screen is that for automatic residual cutting data for bar machining.

#9208 1 : Program detail data screen for automatic residual cutting data

0 : Program detail data screen for other than automatic residual cutting data

(4) Screen control

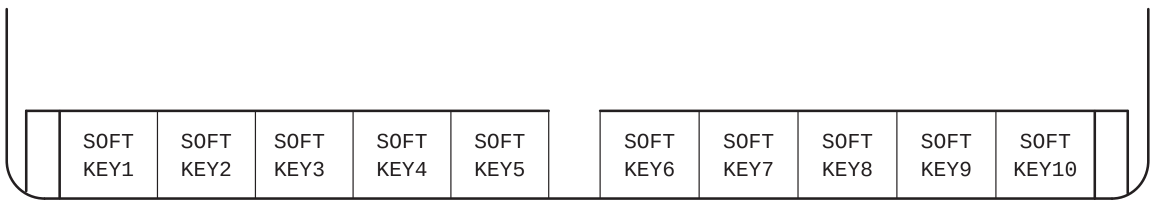
Use the right-hand area in the figure below to dispaly guidance. When the guidance is diaplayed upon cursor movement, the FANUC system program clears the area and calls the user program. The user program need not clear the area.

```
              1          2          3          4          5          6          7
    0123456789 0123456789 0123456789 0123456789 0123456789 0123456789 0123456789 0123456789
 0  EDIT THE PROCESSING PROGRAM  ***  PROCES (01) ROUGH  ***                        [CAP]
 1
 2  MACHINING     P=         BAR
 3  AREA          A=   OUTER END
 4  HEAD          H=     HEAD- L
 5  PASSPOINT1   X1=    135. 000
 6               Z1=     55. 000
 7  PASSPOINT2   X2=    130. 000
 8               Z2=     50. 000
 9  START PNT.   CX=    105. 000
10               CZ=      5. 000
 1  ESCAPE AMNT  EA=      3. 200
 2  S-DRCT.      RS=      NORMAL
 3  FINISHING    FX=      0. 500
 4               FZ=      0. 300
 5  PROC MOVE.   MP=    STANDARD
 6  SPINDLGEAR   GM=       AUTO.
 7  COOLANT      CM=          ON
 8
 9
20
 1
 2
 3
 4
```

## E.6.8 Displaying a Window by Using a User Program and Pressing a Soft Key on the Program Process Data Screen (Not Supported by Ver.1.)

### E.6.8.1 Overview

FANUC Super CAP T/CAP II T allow a user-created program to draw a user's own window while a machining program is being edited. The window is drawn when [SOFT KEY 10] is pressed on the program process data screen.




### E.6.8.2 Details

(1) Displaying a window by pressing a soft key

(a) Number of the program used to display a window, and starting that program

O1025 is assigned to the program used to display a window upon pressing a soft key. User program O1025 is called in each cycle of a system task if [SOFT KEY 10] is pressed while a machining program is being edited on the program process data screen.

To close the window, set a value of 99 in system variable #9120 of user program O1025. The system closes the window and does not call the user program in the subsequent cycles.

(b) Displaying the window and monitoring keys

The system erases the soft key section, while leaving the remainder of the screen as is. The user program shall open a window and create a drawing in that window. Enter the character string for each soft key to be used with the user program, in the corresponding frame, using G243 or another suitable means.

Input by soft keys, page/cursor keys, numeric keys, and other keys can be monitored using system variables #8501 and #8503.

(2) Displaying the character string for the soft key used to display a window

(a) Registering soft key character strings

To display a window, press [SOFT KEY 10]. Register the character string for the soft key in the selected display language, using the following format. Up to six half-size characters can be registered for each row. Up to 49 character strings can be registered.

```
O119*                                                    * : Selected language
N9102 ('123456')  ;  (Soft key  Upper row)      =1 : English
N9103 ('7890AB')  ;  (Soft key 2  Lower row)   =2 : Japanese
N9104 ('        ')  ;  (Soft key 3  Upper row)   =3 : German
N9105 ('        ')  ;  (Soft key 3  Lower row)   =4 : French
   ⁓                                                     =5 : Italian
N9198 ('  End  ')  ;  (Soft key 50  Upper row) =6 : Spanish
N9199 ('        ')  ;  (Soft key 50  Lower row) =7 : Chinese
                                    ↑                   =8 : Korean
              Soft key character string number (#9209)
```

(b) System variable for displaying the soft key

#9209 : System variable indicating the character string to be displayed on [SOFT KEY 10] when a window is displayed by pressing that soft key

Set the corresponding number, between 2 and 50, for the character string to be displayed on [SOFT KEY 10]. The system displays the corresponding character string.

(c) Setting a value in system variable #9209

Process data screen : Set the value, using user program O1022 for displaying a window, depending on the position of the cursor.

Detailed data screen : Set the value, using user program O1023 to display graphic guidance.

## E.6.9 Displaying a Window by Using a Soft Key on the Tooling Data Screen (Not Supported by Ver.1.)

### E.6.9.1 Overview

FANUC Super CAP T/CAP II T allow a user-created program to draw the user's own window when a soft key is pressed on the tooling data screen.

### E.6.9.2 Setting a parameter

| Parameter | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 9779 | | | | USRTLF | USRTLP | | | |

bit3 (USRTLP)
0 : Displays the standard FANUC setup.(*)
1 : Calls the user program (O1019) to display a window on the tooling data screen.
* The standard FANUC setup is not provided for the T two-spindle type or the TT one-spindle type.

bit4 (USRTLF)
0 : Does not call the user program (O1018) to display a window on the tooling data screen.
1 : Calls the user program (O1018) to display a window on the tooling data screen.

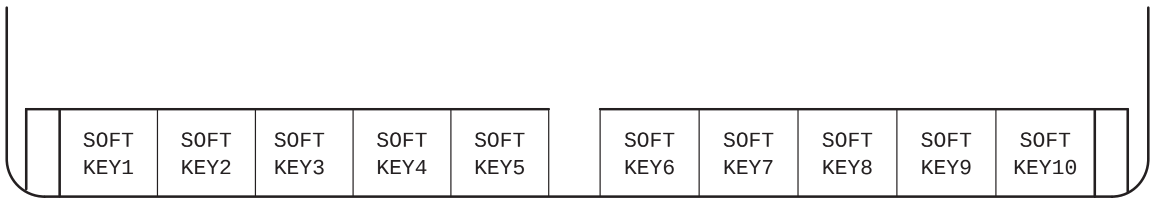
## E.6.9.3 Details

(1) Displaying a window on the tooling data screen

(a) Number of the program used to display a window, and starting that program

O1018 and O1019 are assigned to the programs that display a desired window upon a soft key being pressed. User programs O1018 and O1019 are called in each cycle of a system task when [SOFT KEY 7] and [SOFT KEY 6] are pressed, respectively.

To close the window, set a value of 99 in system variable #9120. The system closes the window and does not call the user program in subsequent cycles.




(b) Displaying the window and monitoring keys

The system erases the soft key section while leaving the remainder of the screen as is. The user program shall open a window and create a drawing in that window. Enter the character string for each soft key to be used with the user program, in the corresponding frame, using G243 or another suitable means.

Input by soft keys, page/cursor keys, numeric keys, and other keys can be monitored using system variables #8501 and #8503.

(c) System variables

i) System variables indicating the position of the cursor on the screen: #9108, #9109
In system variables #9108 and #9109, values indicating the position of the cursor on the screen are set in character coordinates.

#9108 : X coordinate of the cursor position
#9109 : Y coordinate of the cursor position

ii) System variable indicating the ID number of the tool used in the process indicated by the cursor: #9182
The variable is set to the tool ID number of the process indicated by the cursor.

iii) System variable indicating the tool post having the tool to be used in the process, and indicated by the cursor : #9942

#9942 0 or 1 : Tool of head 1 of TT or T (two-spindle type), tool of T (one-spindle type)
2: Tool of head 2 of TT or T (two-spindle type)

(2) Displaying the character string for the soft key used to display a window on the tooling data screen

(a) Registering a soft key character string

To display a window on the tooling data screen, press [SOFT KEY 6] or [SOFT KEY 7]. Register the soft key character string in the selected display language, using the following format. Up to 12 half-size characters can be registered.

```
O119*                                        * : Selected language
N9995 ('123456123456') ; (Soft key7)             =1 : English
N9996 ('123456123456') ; (Soft key6)             =2 : Japanese
                                                 =3 : German
                                                 =4 : French
                                                 =5 : Italian
                                                 =6 : Spanish
                                                 =7 : Chinese
                                                 =8 : Korean
```

## E.6.10 Starting the Chuck/Tailstock Barrier Setting Screen (Not Supported by Ver.1.)

### E.6.10.1 Overview

FANUC Super CAP T/CAP II T support parameter setting that enables a user-created program to use the chuck/tailstock barrier function of the NC. This is done by pressing a soft key on the presetting menu. This function uses the optional chuck/tailstock barrier function of the NC.

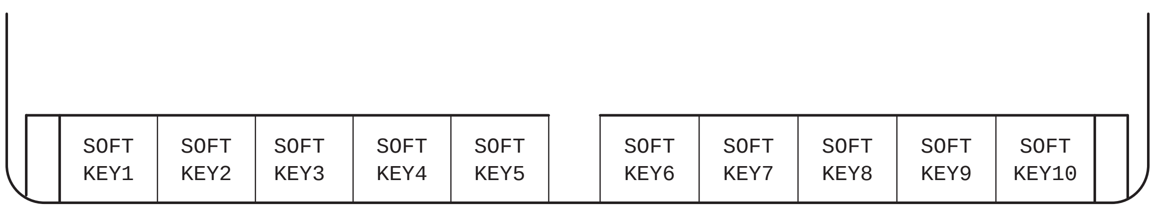
### E.6.10.2 Details

(1) Chuck/tailstock barrier setting screen

(a) Number of the program used to display the chuck/tailstock barrier setting screen, and starting that program

O1026 and O1027 are assigned to the programs that display the chuck/tailstock barrier setting screen. If the cursor is positioned to CHUCK NO. or CHUCK BARRIER of the presetting menu, user program O1026 is called. If the cursor is positioned to TAILSTOCK NO. or TAILSTOCK BARRIER on the same screen, user program O1027 is called. The program is called in each cycle of a system task when [SOFT KEY 9] is pressed.

To close the window, set a value of 99 in system parameter #9120. The system closes the window and does not call the user program in subsequent cycles.




(b) Displaying the screen and monitoring keys

The system erases the soft key section while leaving the remainder of the screen as is. The user program shall open the window and create a drawing in that window. Enter the character string for each soft key to be used with the user program, in the corresponding frame, using G243 or another suitable means.

Input by soft keys, page/cursor keys, numeric keys, and other keys can be monitored using system variables #8501 and #8503.

(2) Displaying the character string for the soft key used to display the chuck/tailstock barrier setting screen

(a) Registering a soft key character string

To display the chuck/tailstock barrier setting screen, press [SOFT KEY 9]. Register the soft key character string in the selected display language, using the following format. Up to twelve half-size characters can be registered.

```
O119*                                              * : Selected language
N9993 ('123456123456') ;
                    Soft key string for the chuck  =1 : English
                    barrier setting screen         =2 : Japanese
N9994 ('123456123456') ;
                    Soft key string for the        =3 : German
                    tailstock setting screen       =4 : French
                                                   =5 : Italian
                                                   =6 : Spanish
                                                   =7 : Chinese
                                                   =8 : Korean
```

# E.7 SETTING PARAMETERS USED FOR COMPILER

## E.7.1 Overview

FANUC Super CAP T/CAP II T allows the machine tool builder to create his own unique system. Programs coded by the machine tool builder and software offered by FANUC can be combined and stored in the ROM module.

The procedure for combining and storing the programs in the ROM module is the same as that for compiling macro-executor programs and P-code programs in the standard format and storing them in the ROM module. However, some restrictions apply to some parameters used for compiling programs with the special macro-executor.

The restrictions that apply to parameters used for compiling are described below.

## E.7.2 Setting Parameters Used for Compiling

| Compile parameter | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 9000 | LD6 | M3MB | M2MB | | | | | |

#7 (LD6) 1 : Be sure to set this bit to 1.

| Compile parameter | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 9001 | | | | | | | LOC4 | SEQN |

#0 (SEQN) 0: The sequence number is not output to a P-code program.
1: The sequence number is output to a P-code program.
★ Be sure to set this bit to 1.

#1 (LOC4) 0: Be sure to set this bit to 0.

| Compile parameter | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 9002 | EXIT1 | PWSR | | | EVF | | | |

#3 (EVF) 0: Extended special P-code variables No. 20000 and after are of the floating point type.
1: Extended special P-code variables No. 20000 and after are of the integer type.
★ Be sure to set this bit to 0.

#6 (PWSR) 0: The function for searching for a P-code workpiece number is disabled.
1: The function for searching for a P-code workpiece number is enabled.
★ Be sure to set this bit to 1.

#7 (EXT1) 0: The extended function is not used.
1: The extended function is used.
★ Be sure to set this bit to 1.

| Compile parameter | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 9003 | TCF | | | | | | | |

#7 (TCF) 0: The special function for entering conversational programs is not used.
1: The function for entering conversational programs is used.
★ Be sure to set this bit to 1.

Compile parameter

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 9007 | MFOPT | TFOPT | | | | TTVR2 | | |

#2 (TTVR2) 0: Extended special P-code variables No. 20000 and after are separated for head 1 and head 2.

1: Extended special P-code variables No. 20000 and after are shared by head 1 and head 2.

★ Be sure to set this bit to 0 when Super CAP T Ver. 1 is used for the Series 16-TTA.

#6 (TFOPT) 0: The conversational programming function for the Series 16-T/TTA is not used.

1: The conversational programming function for the Series 16-T/TTA is used.

★ Be sure to set this bit to 1.

#7 (MFOPT) 0: The conversational programming function for the Series 16-MA is not used.

1: The conversational programming function for the Series 16-MA is used.

★ Be sure to set this bit to 0.

Compile parameter

| | |
|---|---|
| 9021 | G code (1 to 999) which calls the P-code program No. O9018 |

The macro program used in the auxiliary process is assigned P-code program No. O9018 in FANUC Super CAP.

Compile parameter

| | |
|---|---|
| 9022 | G code (1 to 999) which calls the P-code program No. O9019 |

The macro program used in the auxiliary process is assigned P-code program No. O9010 in FANUC Super CAP.

Compile parameter

| | |
|---|---|
| 9037 | Number of special P-code variables No. 10000 and after to be used |

★ Be sure to set this bit to 7.

**NOTE**
The special P-CODE variables of FANUC Super CAP T are usually set in system variables #10000 to #10699. They can be extended to #10000 to #13999 if desired.

Compile parameter

| | |
|---|---|
| 9044 | Number of special P-code variables No. 20000 and after to be used |

★ Be sure to set this parameter to 2044.

**NOTE**
The extended special P-CODE variables of FANUC Super CAP T are used by the system. The user program cannot use these variables.

# E.8 COMMAND USED IN THE DYNAMIC GRAPHIC FUNCTION

## E.8.1 Outline

To perform animated simulation in the dynamic graphic function (A02B-0121-J973), data such as the figures of blanks, the chuck, tailstock, and tools are required to be sent using commands [G10 P90???...] in advance from the NC unit to the graphic section. A program consisting of the commands [G10 P90??? ...] needs to be operated before a machining program coded in the NC format operates to display its animated simulation.

When the conversational programming function is used, the necessary data is automatically sent to the graphic section.

When either the menu or the tool is changed, execute the program consisting of the commands [G10 P90???...] to send the data to the graphic section.

## E.8.2 Description

(1) Start point for animated simulation

Command format    G10 P90000 X_Z_ ;

Specifies the coordinates of the start point.

> **NOTE**
> When commands are issued in the following order animated simulation does not start, but the data is registered in the graphic section: A registration command, G10 P90051, G10 P90052, G10 P90053, G10 P90054, or G10 P90055, this command, and the registration end command G10 P90050

(2) Drawing a straight line

Command format    G10 P90001 X_Z_ ;

Draws a straight line from the current position to the specified end point.

> **NOTE**
> When commands are issued in the following order, animated simulation does not start, but the data is registered in the graphic section: A registration command, G10 P90051, G10 P90052, G10 P90053, G10 P90054, or G10 P90055, this command, and the registration end command G10 P90050.

(3) Drawing an arc

Command format

G10 P90002 X_Z_I_K_ ; for drawing clockwise
G10 P90003 X_Z_I_K_ ; for drawing counterclockwise

Draws an arc around the specified center of the arc from the current position to the specified end point.

X : X coordinate of the end point
Z : Z coordinate of the end point
I : X coordinate of the center of the arc
K : Z coordinate of the center of the arc

**NOTE**
When commands are issued in the following order, animated simulation does not start, but the data is registered in the graphic section: A registration command, G10 P90051, G10 P90052, G10 P90053, G10 P90054, or G10 P90055, this command, and the registration end command G10 P90050.

(4) Filling

Command format G10 P90004 ;

Fills an area.

**NOTE**
The color command G10 P90006 specifies a color used for filling an area and the start point command G10 P90001 for animated simulation specifies the center of the area.

(5) Type of line

Command format G10 P90005 Q_ ;

Specifies the type of line.

Q= 0 : Solid line
1 : Dotted line
2 : Line with alternate dots and long segments
3 : Line with alternate long segments and sets of two dots
4 : Deletion

**NOTE**
The length of each line is specified.

(6) Color

Command format G10 P90006 Q_ ;

Specifies the color.

| | |
|---|---|
| Q= 0 : Black | 4 : Blue |
| 1 : Red | 5 : Pink |
| 2 : Green | 6 : Light blue |
| 3 : Yellow | 7 : White |

(7) End of registration of the figures of tools, workpieces, the chuck, and tailstock

Command format G10 P90050 ;

Be sure to specify this command when the figure data of tools, workpieces, the chuck, and tailstock have been registered.

(8) Start of tool-figure registration

Command format G10 P90051 X_Z_Q_ ;

Specifies the start of tool-figure registration as well as the position of the tip of the tool and the tool registration number on the graphic screen.

X : X coordinate of the tip of the tool
Z : Z coordinate of the tip of the tool
Q : Tool registration number on the graphic section (1 to 16)

**NOTE**
The start point command G10 P90000 for animated simulation, command G10 P90001 for drawing a straight line, and command G10 P90002 or G10 P90003 for drawing an arc specify the figure of a tool. The figure of the tip of a tool must be closed.

(9) Start of registering the chuck profile

Command format G10 P90052 X_Z_ ;

Specifies the start of registering the chuck figure as well as the start point of the figure.

X : X coordinate of the start point for the chuck profile
Z : Z coordinate of the start point for the chuck profile

**NOTE**
The start point command G10 P90000 for animated simulation, command G10 P90001 for drawing a straight line, and command G10 P90002 or G10 P90003 for drawing an arc specify the figure of the chuck. The figure must be closed.

(10) Start of tailstock-figure registration

Command format G10 P90053 X_Z_ ;

Specifies the start of registering the tailstock figure as well as the start point of the figure.

X : X coordinate of the start point for the tailstock profile
Z : Z coordinate of the start point for the tailstock profile

**NOTE**
The start point command G10 P90000 for animated simulation, command G10 P90001 for drawing a straight line, and command G10 P90002 or G10 P90003 for drawing an arc specify the tailstock figure. The figure must be closed.

(11) Start of registering a blank profile

Command format G10 P90054 X_Z_ ;

Specifies the start of registering the blank profile as well as the start point of the figure.

X : X coordinate of the start point for the blank profile
Z : Z coordinate of the start point for the blank profile

**NOTE**

The start point command G10 P90000 for animated simulation, command G10 P90001 for drawing a straight line, and command G10 P90002 or G10 P90003 for drawing an arc specify the blank figure. The figure must be closed.

(12) Start of the registration of a blank profile viewed from the end face

Command format G10 P90055 X_Z0. ;

Specifies the start of the registration of a blank figure viewed from the end face as well as the diameter.

X : Diameter of the blank

(13) Registering the profile of the tool used for C-axis machining

Command format G10 P90056 X_Z_Q_R_ ;

Specifies the diameter, length, and tool type, as well as the registration number in the graphic section.

X : Diameter of the tool
Z : Length of the tool
Q : Registration number of the tool in the graphic section (17 to 32)
R = 0 : For a tool which cuts end faces
1 : For a tool which cuts side faces

(14) Size of an exploded view

Command format G10 P90057 X_Z_C_ ;

Specifies the area (X, Z) for which an exploded view is drawn and the diameter (C) of a workpiece.

X : Z coordinate of the start point
Z : Z coordinate of the end point
C : Diameter of a workpiece

(15) Selecting the tool used in tool post 1

Command format G10 P90060 Q_R_ ;

Selects the tool to be used in tool post 1 and specifies the registration number of the tool in the graphic section.

Q : Registration number of the tool (1 to 16) in the graphic section

R : Axes to which the mirror function is applied to generate the figure of the tool
= 0 : X-axis off, Z-axis off
= 1 : X-axis on, Z-axis off
= 2 : X-axis off, Z-axis on
= 3 : X-axis on, Z-axis on

**NOTE**

When the conversational programming function is used in FANUC Super CAP, the registration number of the tool in the graphic section can be read from macro variable No. 20705 or 20704 (variable No. 20704 is used for the tool used for residual machining in bar machining).

(16) Selecting the tool used in tool post 2

Command format G10 P90061 Q_R_ ;

Selects the tool to be used in tool post 2 and specifies the registration number of the tool in the graphic section.

Q : Registration number (1 to 16) of the tool in the graphic section

R : Axes to which the mirror function is applied to generate the figure of the tool
= 0 : X-axis off, Z-axis off
= 1 : X-axis on, Z-axis off
= 2 : X-axis off, Z-axis on
= 3 : X-axis on, Z-axis on

**NOTE**

When the conversational programming function is used in FANUC Super CAP, the registration number of the tool in the graphic section can be read from macro variable No. 20705 or 20704 (variable No. 20704 is used for the tool used for residual machining in bar machining).
This command is effective only for a lathe with one spindle and two turrets.

(17) Selecting the tool used for C-axis machining

Command format G10 P90062 Q_R_ ;

Selects the tool to be used for C-axis machining and specifies the registration number of the tool in the graphic section.

Q : Registration number (17 to 32) of the tool in the graphic section

R : Color used for drawing the tool

Q= 0 : Black
16 : Red
32 : Green
48 : Yellow
64 : Blue
80 : Pink
96 : Light blue
112 : White

**NOTE**

When the conversational programming function is used in FANUC Super CAP, the registration number of the tool in the graphic section can be read from macro variable No. 20705.

(18) Animation screen

Command format G10 P90070 Q_ ;

Specifies the type of drawing used for animated simulation.

Q= 0 : Drawing used for turning
1 : End-face drawing used for C-axis machining
2 : Exploded view used for C-axis machining

(19) Deleting figures on the screen

Command format G10 P90099 ;

Deletes the figures on the screen.

(20) Start point for drawing the chuck figure

Command format G10 P90100 X_Z_ ;

Specifies the start point for drawing the profile of the chuck.

X : X coordinate of the start point for drawing the profile of the chuck in the current workpiece coordinate system

Z : Z coordinate of the start point for drawing the profile of the chuck in the current workpiece coordinate system

(21) Start point for drawing the profile of the tailstock

Command format G10 P90101 X_Z_ ;

Specifies the start point for drawing the profile of the tailstock.

X : X coordinate of the start point for drawing the profile of the tailstock in the current workpiece coordinate system

Z : Z coordinate of the start point for drawing the profile of the tailstock in the current workpiece coordinate system

(22) Start point for drawing the profile of a blank

Command format G10 P90102 X_Z_ ;

Specifies the start point for drawing the profile of a blank.

X : X coordinate of the start point for drawing the profile of a blank in the current workpiece coordinate system

Z : Z coordinate of the start point for drawing the profile of a blank in the current workpiece coordinate system

(23) Drawing the end-face view used for C-axis machining

Command format G10 P90103 ;

Draws the end-face view used for C-axis machining.

(24) Drawing the exploded view used for C-axis machining

Command format G10 P90104 ;

Draws the exploded view used for C-axis machining.

(25) Copying the profile of a workpiece

Command format G10 P90200 X_Z_I_K_ ;

Copies the profile of a workpiece drawn in the specified area from the first graphic screen to the second graphic screen or vice versa.

X : X coordinate of the upper right point of the area to be copied

Z : Z coordinate of the upper right point of the area to be copied

I : X coordinate of the lower left point of the area to be copied

K : Z coordinate of the lower left point of the area to be copied

**NOTE**

The command copies the profile of a workpiece to the screen on which the command is issued. When the command is specified in head 2, for example, the profile is copied from head 1 to head 2. The coordinates must be those on the source screen. When the profile is copied from head 1 to head 2, for example, the coordinates must be those in the workpiece coordinate system of head 1. The coordinates cannot be omitted.
This command is effective only for a lathe with two spindles and two turrets.

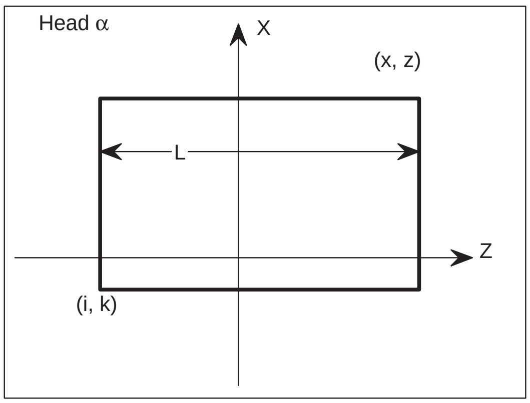
(26) Coordinate conversion to those in the copy destination

Command format G10 P90201 X_Z_ ;

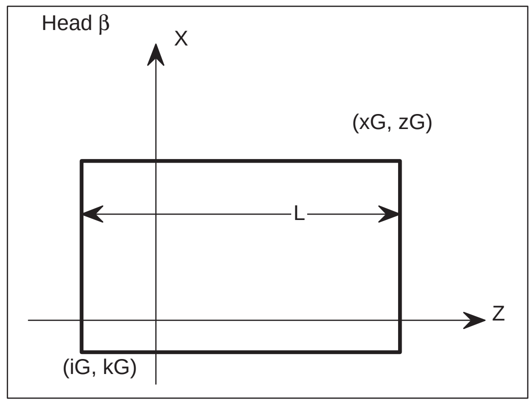
X, Z : Differences between coordinates used in two heads with reference to those in the source area

**NOTE**

1 If the command G10 P90201 is not specified before the command G10 P90200 is specified, the differences between the coordinates used in two heads are assumed to be 0. Command G10 P90201 is a continuous-state command. The differences between coordinates specified by this command is not reset until this command is specified next.
When the directions of two axes are the same between two coordinate systems as shown below, to copy the rectangular area specified by the two points (x, z) and (i, k) in head α to that specified by the two points (x', z') and (i', K') in head β, specify the following:




Coordinates of the center point of the screen: $(x_0, z_0)$

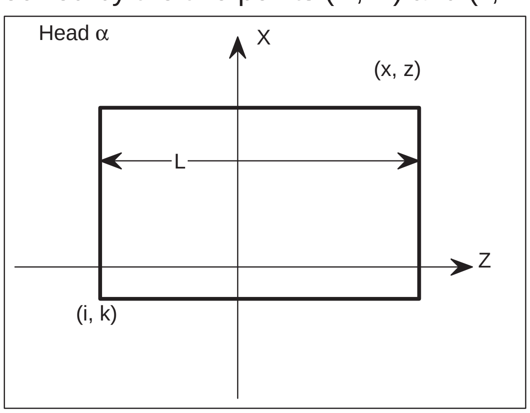



Coordinates of the center point of the screen: $(x_0', z_0')$

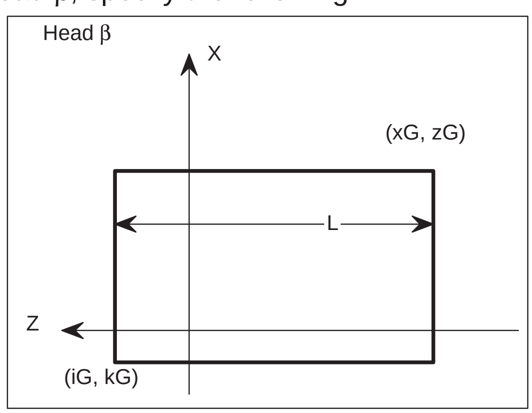
G10 P90201 X((x' – x) + (–$x_0$' + $x_0$) Z((z' – z) + (–$z_0$' + $z_0$); (can be omitted)
G10 P90200 Xx Zz Ii Kk;

Specify these commands in head β.

2 When the Z-axes in the two coordinate systems have the same directions as shown above, the command cannot be used in FANUC Super CAP.
When the horizontal Z-axes in the two coordinate systems have opposite directions as shown below, to copy the rectangular area specified by the two points (x, z) and (i, k) in head α to that specified by the two points (x', z') and (i', K') in head β, specify the following:




Coordinates of the center point of the screen: $(x_0, z_0)$




Coordinates of the center point of the screen: $(x_0', z_0')$

G10 P90201 X((x' – x) + ($x_0$' + $x_0$) Z((–z' – z) + ($z_0$' + $z_0$); (can be omitted)
G10 P90200 Xx Zz Ii Kk;

Specify these commands in head β.

**NOTE**

3 When the Z-axes in the two coordinate systems have opposite directions as shown above, the command can be used in FANUC Super CAP.
The signs of coordinates used in the command G10 P90201 are as follows:
For the Z coordinates:
Z (+/–A –+B –/+C +/–D) when the direction of the Z-axis is right/left
where,
A : Z coordinate of the rightmost point in the destination
B : Z coordinate of the rightmost point in the source
C : Z coordinate of the center point of the screen in the destination
D : Z coordinate of the center point of the screen in the source
The upper right point is not affected by the direction of the coordinate system.
For the X coordinates:
X (+/–A –/+B –/+C +/–D) when the direction of the X-axis is up/down
where,
A : X coordinate of the uppermost point in the destination
B : X coordinate of the uppermost point in the source
C : X coordinate of the center point of the screen in the destination
D : X coordinate of the center point of the screen in the source
When the X-axis is a horizontal axis and the Z-axis is a vertical axis, the same signs can be used by interchanging the X- and Z-axes.

4 A user program cannot refer to the coordinates of the center point of the screen in FANUC Super CAP.

(27) Switching the display mode

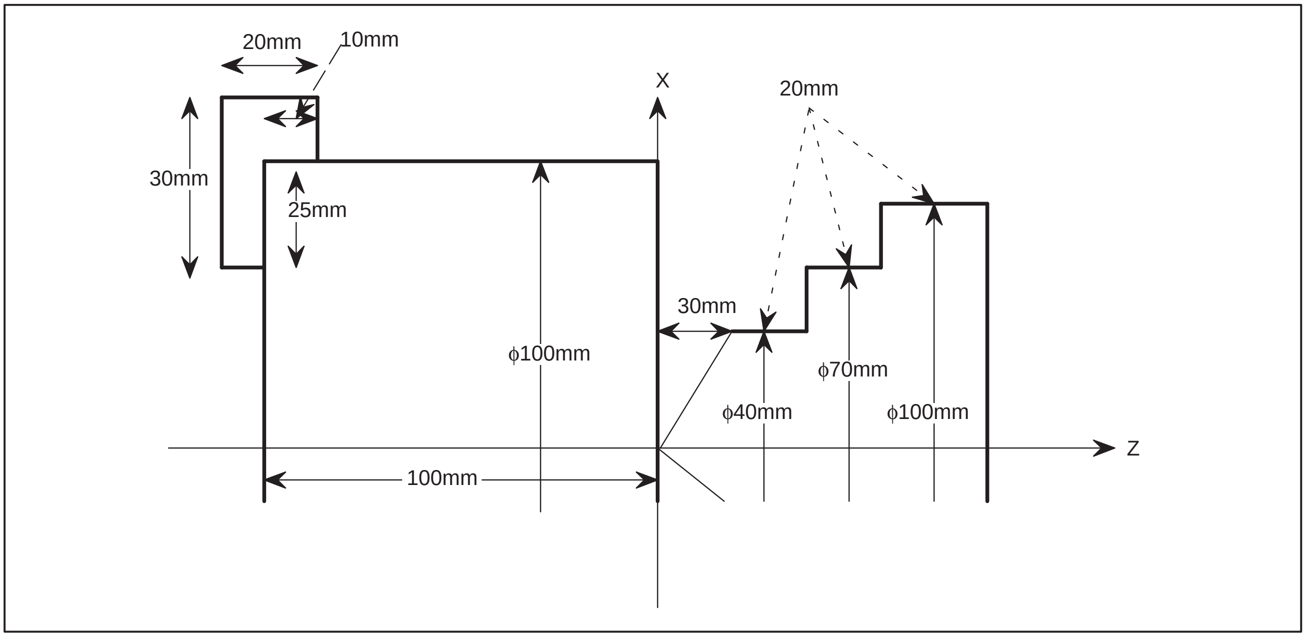
Command format G10 P90210 Q_ ;

Switches between the two-spindle and one-spindle display modes.

**NOTE**

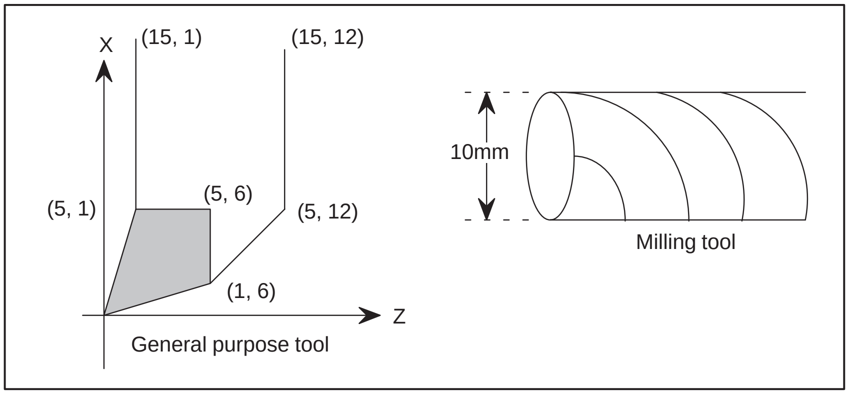
Issuing this command changes the mode and initializes the screen. This command is effective only for a lathe with two spindles and two turrets.

## E.8.3 Example




The following program registers the profiles of a workpiece, the chuck, and tailstock in the graphic section.

```
G10  P90099 ;                  ----  Screen erase
G10  P90055 X100.  Z0.  ;
G10  P90050 ;
G10  P90054 X0.  Z-100.  ;
G10  P90001 X100.  Z-100.  ;
G10  P90001 X100.  Z0.  ;            Blank
G10  P90001 X0.  Z0.  ;
G10  P90001 X0.  Z-100.  ;
G10  P90050 ;                  ----  End of registration
G10  P90102 X0.  Z0.  ;        ----  Blank standard position
G10  P90052 X50.  Z-100.  ;
G10  P90001 X110.  Z-110.  ;
G10  P90001 X110.  Z-90.  ;
G10  P90001 X100.  Z-90.  ;          Chuck profile
G10  P90001 X100.  Z-100.  ;
G10  P90001 X50.  Z-100.  ;
G10  P90001 X50.  Z-100.  ;
G10  P90050 ;                  ----  End of registration
G10  P90100 X0.  Z0.  ;        ----  Chuck standard position
G10  P90053 X0.  Z10000.  ;
G10  P90001 X40.  Z10020.  ;
G10  P90001 X40.  Z10030.  ;
G10  P90001 X70.  Z10030.  ;
G10  P90001 X70.  Z10050.  ;         Tail stock data
G10  P90001 X100.  Z10050.  ;
G10  P90001 X100.  Z10070.  ;
G10  P90001 X0.  Z10070.  ;
G10  P90001 X0.  Z10000.  ;
G10  P90050 ;                  ----  End of registration
G10  P90101 X0.  Z-10000.  ;         Tail stock standard position
(G10  P90101  X0.  Z10000.  ;)
```




The following program registers the above tools.

```
G10  P90099 ;                  ----  Screen erase
G10  P90051 X0.  Z0.  Q1  ;
G10  P90001 X5.  Z1.  ;
G10  P90001 X5.  Z6.  ;              Chip figure
G10  P90001 X1.  Z6.  ;                                Tool figure
G10  P90001 X0.  Z0.  ;
G10  P90000 X15.  Z1.  ;
G10  P90001 X5.  Z1.  ;
G10  P90001 X0.  Z0.  ;              Chunk profile
G10  P90001 X1.  Z6.  ;
G10  P90001 X5.  Z12.  ;
G10  P90001 X15.  Z12.  ;
G10  P90050 ;                  ----  End of registration

G10  P90056  X10.  Z40.  Q17  R0. ; Registering milling tool shape
```

## E.8.4 Transferring a Workpiece in Animated Simulation

In a lathe with two spindles, it is necessary to copy the profile of a workpiece between heads using the following commands to display the transfer process in animated simulation.

G10 P90201 X_Z_ ;

G10 P90200 X_Z_I_K_ ;

However, command G10 P90201 cannot be specified without the coordinates of the center point of the screen. The following data is assigned to P-code variable No. 20623 when the conversational programming function is used in FANUC Super CAP.

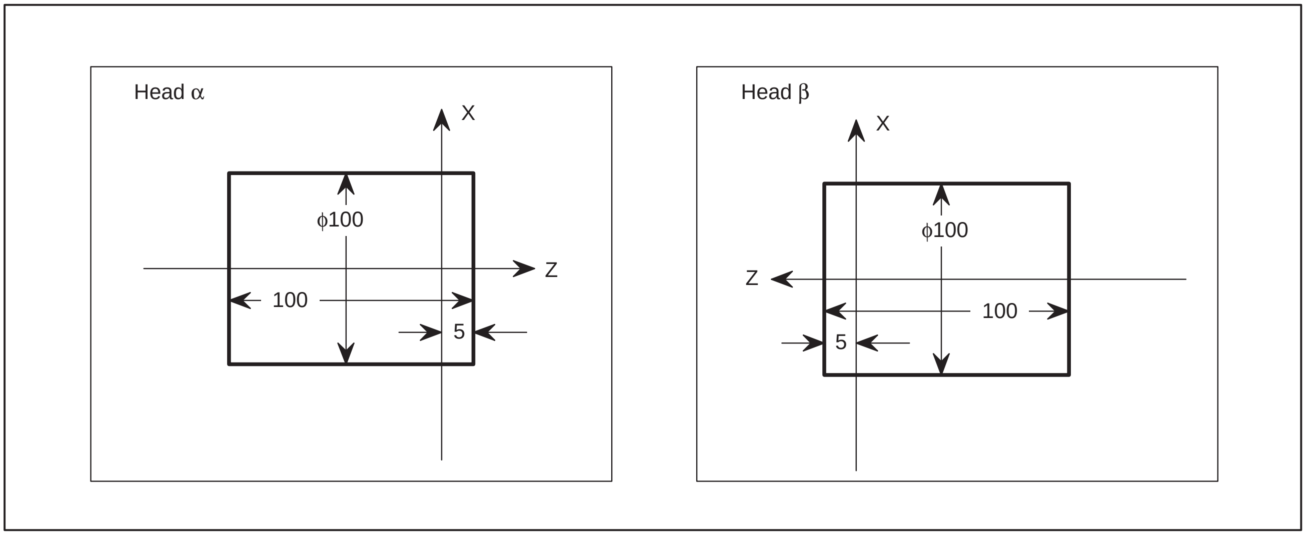
- When the program origin is positioned on the end face of a workpiece
  #20623 = A + B + C
- When the program origin is positioned on the end face of the chuck
  #20623 = A + B – C

where,

A : Z coordinate of the center point of the screen in head 1
B : Z coordinate of the center point of the screen in head 2
C : Length of the product

(1) Example

When the parameter No. 6510 for head 1 is 14 and that for head 2 is 34 :




In this example, the X coordinates of the center points are the same in both screens. Specify as follows in head β :

G10 P90201 X0. Z(–(–95) –5 +$Z_0$' +$Z_0$). ;
G10 P90200 X100. Z5. I-100. K-95. ;
(when diameter programming is used for the X-axis)

> **NOTE**
> In FANUC Super CAP T/CAP II T, parameter No. 6510 for head 1 is set to 1, 4, 11, or 14 and that for head 2 is set to the value set in parameter No. 6510 for head 1 plus 20. So the X coordinates of the center points are the same in both screens.

# E.9 COMMANDS FOR READING/WRITING VARIABLES STORED IN THE REMOTE HEAD

## E.9.1 Outline

In FANUC Super CAP T/CAP II T, all programs are stored in head 1. Extended P-code variables No. 20000 and after must be separated for head 1 and head 2. When head 2 is selected, user programs in head 2 are required to read/write extended P-code variables for head 1. These commands have been added for reading/writing variables stored in the remote head.

## E.9.2 Description

G316 Pp Dd ; (1)
G316 Dd Qq ; (2)

(1) This command reads the value of parameter No. d stored in the remote head and stores it in parameter No. p in the local head.

(Example) G316 P10000 D20000 ;
The value of parameter No. 20000 in the remote head is transferred to parameter No. 10000 in the local head.

(2) This command writes the value of parameter No. q stored in the local head in parameter No. d stored in the remote head.

(Example) G316 D10001 Q20001 ;
The value of parameter No. 20001 in the local head is transferred to parameter No. 10001 in the remote head.

(Restrictions)

- This command cannot be used in an execution macro.
- System variables #9000 to #9999 cannot be specified.
- In a conversational macro program, variable d of the remote head must be a special P-CODE variable (#10000 or later), extended special P-CODE variable (#20000 or later), or common variable.
- In an auxiliary macro program, variable d of the remote head must be a special P-CODE variable (#10000 or later) or extended special P-CODE variable (#20000 or later).

# E.10 FUNCTION FOR CONVERTING PROGRAMS USED FOR THE AUXILIARY AND TRANSFER PROCESSES TO NC STATEMENTS

## E.10.1 Outline

When programs used for the auxiliary and transfer processes are converted to NC statements, the machine conditions strongly affect the NC statements even if the same data is used. Super CAP T/CAP II T provides two types of NC statements to be converted from programs used for the auxiliary and transfer processes. They are NC statements which represent the programs to be executed and those only for calling sub-programs.

It is also possible to select whether programs are called either from the ROM or from the RAM.

## E.10.2 Setting Parameters

| Parameter | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 9777 | | NCSNTF | | | | | | |

Bit 6 (NCSNTF) 0 : NC statements which represent programs to be executed are output for the auxiliary and transfer processes.

1 : NC statements only for calling sub-programs are output for the auxiliary and transfer processes. (For details, see 6.8.3.)

| Parameter | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 9778 | | | | | | | AUX | TRANS |

Bit 0 (TRANS) 0 : Sub-programs are called from the ROM in the transfer process.

1 : Sub-programs are called from the RAM in the transfer process.

Bit 1 (AUX) 0 : Sub-programs are called from the ROM in the auxiliary process.

1 : Sub-programs are called from the RAM in the auxiliary process.

## E.10.3 Format Used for NC Statements Only for Calling Sub-programs

When bit 6 of parameter 9777 is 1, programs for calling sub-programs in the auxiliary and transfer processes are converted to NC statements in the following format.

(1) When sub-programs are called from the ROM (when bit 0 of parameter No. 9778 is set to 0 for the transfer process, and/or when bit 1 of parameter No. 9778 is set to 0 for the auxiliary process)

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| G | (1) | A | (+7) | B | (+9) | C | (+11) | I | (+13) | J | (+15) | K | (+17) |
| | | I | (+19) | J | (+21) | K | (+23) | I | (+25) | J | (+27) | K | (+29) |
| | | I | (+31) | J | (+33) | K | (+35) | I | (+36) | J | (+37) | K | (+38) |
| | | I | (+39) | J | (+40) | K | (+41) | I | (+42) | J | (+43) | K | (+44) |
| | | I | (+45) | J | (+46) | K | (+47) | I | (+48) | J | (+50) | K | (+51) |
| | | I | (#20500) | | | J | (#20501) | | | K | (#20502) | | |

(1) : For the transfer process, the value of compile parameter No. 9021 is output. For the auxiliary process, the value of compile parameter No. 9022 is output.

(+?) : +? indicates the offset from the start of the process data. The values corresponding to offset +? are output. For details of the process offset data, see 6.2.2 (b) (i).

(#?) : #? indicates the number of a macro variable. The values of the corresponding macro variables are output.

(2) When sub-programs are called from the RAM

- When bit 0 of parameter No. 9778 is set to 1 for the transfer process, M98 P9018;
- When bit 1 of parameter 9778 is set to 1 for the auxiliary process, M98 P9019;

## E.10.4 Note

- If no value is specified for a certain data item shown in 6.8.3.(1), the value 0 is output for the item.
- Version 3 enables NC statement conversion to another format. For details, refer to the FANUC Super CAP T Operator’s Manual (Series 16-TB).

## E.11 IMPROVEMENTS FEATURED BY VERSION 4

(1) The system capacity has been expanded.

Super CAP T Version 3 : 2M bytes

Super CAP T Version 4 : 3M bytes

(2) The macro capacity has been expanded.

Super CAP T Version 3 : 512K bytes

Super CAP T Version 4 : 768K bytes

(3) The link file has been partially modified to reflect the above expansions and to ensure compatibility with the VGA graphics capability.
See the next page.

## E.11.1 Modifications Made to the Link File

To reflect the conversational system capacity and macro capacity expansions and to ensure compatibility with the VGA graphics capability, the link file has been partially changed.

```
/*
/* Conversation MACRO Sample LINK control file.
/*
/*
/* executor file 'FS16TC for VGA'
/*
CNC =c:\mcomp\USR\BH5G_4A.EX1
CNC2 =c:\mcomp\USR\BH5G_4A.EX2
CNC3 =c:\mcomp\USR\BH5G_4A.EX3
PCODE=REAR
/*
/* compile parameter P9000 – P9009 and P9010 – P9059
/*
P9000=11000000
P9001=10000001
P9002=11000000
P9003=10001001
P9007=01000000
P9009=00000100
/*
P9013=200
/*
P9023=320
P9024=321
/*
P9030=27
P9031=28
P9032=29
/*
P9033=97
P9037=7
P9044=2044
/*
P9038=4999
/*
/* Link file's
/*
FILE=CAPS_4A
FILE=BH5G_4A
```

PCODE=REAR: Must not be omitted.

P9000: Set 3M (by setting bit 6 to 1).

| 9000 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | LD6 | R3M | R2M | R1M | R512 | R256 | R128 | VRFY |

R3M = 1: Output to a 3M–byte ROM cassette or ROM module.

P9009 (bits 2 and 1): Number of screen lines used
The higher of these two bits must always be set to 0.
The lower of these two bits can be set arbitrarily by the machine tool builder.

| 9009 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | CM30 | LM30 | |

LM30 = 1: The full 30–line screen is used for VGA display. (Super CAP II T)
LM30 = 0: The full 30–line screen is not used for VGA display. (Super CAP II T)
CM30 = 1: The full 30–line screen is used for VGA display. (CUSTOM)
CM30 = 0: The full 30–line screen is not used for VGA display. (CUSTOM)

**NOTE**

The modifications made to the link file described above are just an example. Set those items other than those indicated by an arrow and comment, as required, as described in this manual.

# E.12 IMPROVEMENTS FEATURED BY FANUC SUPER CAP II T

(1) The system capacity has been expanded.

| | | |
|---|---|---|
| Super CAP T Version 3 | : | 2M bytes |
| Super CAP II T | : | 3M bytes |

(2) The macro capacity has been expanded.

| | | |
|---|---|---|
| Super CAP T Version 3 | : | 512K bytes |
| Super CAP II T | : | 768K bytes |

(3) The software package related to the process data and user program of the macro compiler/executor has been partially modified.
See Section E.12.1.

(4) The software package related to the interface between the system and user programs of the macro compiler/executor has been partially modified.
See Section E.12.2.

(5) The graphics interface of the macro compiler/executor for Super CAP II T has been partially modified for screen display based on VGA graphics.
See Appendix M.

(6) The link file has been partially modified to reflect the expansions above ((1) and (2)) and to ensure compatibility with the VGA graphics capability.
See Section E.12.3.

## E.12.1 Software Package Related to Process Data

### E.12.1.1 Specifications

(a) Data structure of the auxiliary and passing processes

The input format and display format of a data item such as type, head, and display data depends on the data format specified for the data item.

(Data format) = –1:

The data item is not displayed. The cursor cannot be positioned to the item. In the field of the data item, the data for the next data item is moved up and displayed.

(Data format) < 10000:

The data item represents numeric data. The data is entered using the numeric keypad. A value indicated with the data format represents the number of decimal places.

(Data format) ≧ 10000:

The data item is displayed as a string. The data is entered using the soft keys.

(Data format) ≧ 20000:

The data item represents numeric data. The data is entered using the soft keys and numeric keypad.

(Data format) = *ABC*:

When A = 0, the data is displayed in the standard character color.

When A = 1, the data is displayed in reverse video.

When A = 2, an extended palette is used for the character color of B.

When A = 4, an extended palette is used for the background color of C.

(Data format) = *ABC*:

When B = 0, the data is displayed in black
(or in the 8th color when color extension is specified with A).

When B = 1, the data is displayed in color palette 1
(or in the 9th color when color extension is specified with A).

When B = 2, the data is displayed in color palette 2
(or in the 10th color when color extension is specified with A).

When B = 3, the data is displayed in color palette 3
(or in the 11th color when color extension is specified with A).

When B = 4, the data is displayed in color palette 4
(or in the 12th color when color extension is specified with A).

When B = 5, the data is displayed in color palette 5
(or in the 13th color when color extension is specified with A).

When B = 6, the data is displayed in color palette 6
(or in the 14th color when color extension is specified with A).

When B = 7, the data is displayed in color palette 7
(or in the 15th color when color extension is specified with A).

(Data format) = *ABC*:

When C = 0, the background is displayed in color palette 14
(or in the 8th color when color extension is specified with A).

When C = 1, the background is displayed in color palette 1
(or in the 9th color when color extension is specified with A).

When C = 2, the background is displayed in color palette 2
(or in the 10th color when color extension is specified with A).

When C = 3, the background is displayed in color palette 3
(or in the 11th color when color extension is specified with A).

When C = 4, the background is displayed in color palette 4
(or in the 12th color when color extension is specified with A).

When C = 5, the background is displayed in color palette 5
(or in the 13th color when color extension is specified with A).

When C = 6, the background is displayed in color palette 6
(or in the 14th color when color extension is specified with A).

When C = 7, the background is displayed in color palette 7
(or in the 15th color when color extension is specified with A).

(Supplement)

To use extended colors for both characters and their background, or to use reverse video, set the sum of all the states in A:

Example: [12th extended color for characters] + [15th extended color for the background]
A = 2 (character extension) + 4 (background extension)
= 6

Examples of *ABC* specification

Example: [Standard character in black] + [palette 14 for the background] (same as the system data background color)
*ABC* = *000*

Example: [Standard character in extended character palette 12] + [extended palette 15 for the background]
*ABC* = *647*

(Data format) = 100000:

The data item represents a surface speed or rotational speed, and its display and data input are handled in the same way as other processes.

(Data format) = 100001:

The data item represents coolant ON or OFF (1 = ON, 2 = OFF). The data item is displayed as a string. Data can be entered using the soft keys and numeric keypad.

(Data format) = 200000:

The data item is displayed as a string. Data can be entered using the soft keys and numeric keypad.

(Data format) = 300001:

The data item represents a tool ID number. Data can be entered using the soft keys and numeric keypad.

(Data format) = 300002:

The data item represents a T code. Data can be entered using the soft keys and numeric keypad.

## E.12.2 Software Package Related to the Interface Between the System and User Programs

### E.12.2.1 Interface Between the Conversational Program Screen and User Programs

(1) System variables for multi–window display specification

#9111 = X coordinate of the upper–left point of the multi–window

#9112 = Y coordinate of the upper–left point of the multi–window

#9113 = Number of characters in the X direction (horizontal direction of the screen) of the multi–window

#9114 = Number of characters in the Y direction (vertical direction of the screen) of the multi–window

#9115 = Type of frame of the multi–window

0: Thin line, 1: Heavy line, <u>2: Three–dimensional window</u>

#9116 = Color of the frame of the multi–window <u>(color of the title bar when the multi–window is a three–dimensional window)</u>

0: Black, 1: Red, 2: Green, 3: Yellow, 4: Blue, 5: Purple, 6: Light blue, 7: White, 8 to 15: Extended colors

#9120 = Request for display to the system

– When 1 is entered, the system opens a window based on the information of #9111 through #9116. When 2 is entered, the system closes the window. Upon the completion of processing, the system initializes #9120 to 0. Set 99 to return from the detail screen to the process screen. Then, the system displays the process screen.

> **NOTE**
> 1. The multi–window is not displayed when any of the above variables assumes a null value or a value outside the specifiable range.
> 2. When a window is converted to a three–dimensional window, the color of the characters within the window must be changed to a color that is easily legible against the background color of the window.
> 3. When a window is displayed on the macro debugger screen, the color palettes for the conversational screen cannot be used. So, the window may not be displayed in the specified colors in some cases.

#9119 = Request for redisplay to the user program

The system may close a window (for example, for switching to another screen) when no request is issued from a user program. If the system closes a window for the system's reason, the user program must issue another request to display the window. When the user program must reopen a window, the system sets 1. Then, to output a display request to the system, the user program must set the system variable for window display specification. The system initializes #9119 to 0 when the user program performs a read.

Modifications made to the basic menu screen

## E.12.2.2 Specification of Display Characters

(1) Specification of characters to be displayed by the system

(a) Title display

On the basic menu screen, a title specific to each machine tool builder can be displayed.

In N9999 of O1191 to O1198 (for each language), register a title of no more than 50 half–size characters.

<u>Moreover, Super CAP II T allows a title to be displayed using O1030.</u>

<u>When parameter 9975 (MNU) = 0</u>

<u>When O1020 is not linked, only the basic menu screen, which is standard for conversation, is displayed. No title is displayed.</u>

<u>When O1030 is linked, the basic menu screen, which is standard for conversation, is displayed, and the program coded in O1030 is displayed. As with a sample program, a yellow frame, graphic display, characters, and so forth can be displayed as required.</u>

When parameter 9975 (MNU) = 1

When O1020 (O1021) is not linked

The basic menu screen, which is standard for conversation, is not displayed. Because O1020 (O1021) is not linked, no menu is displayed.

When O1020 (O1021) is linked

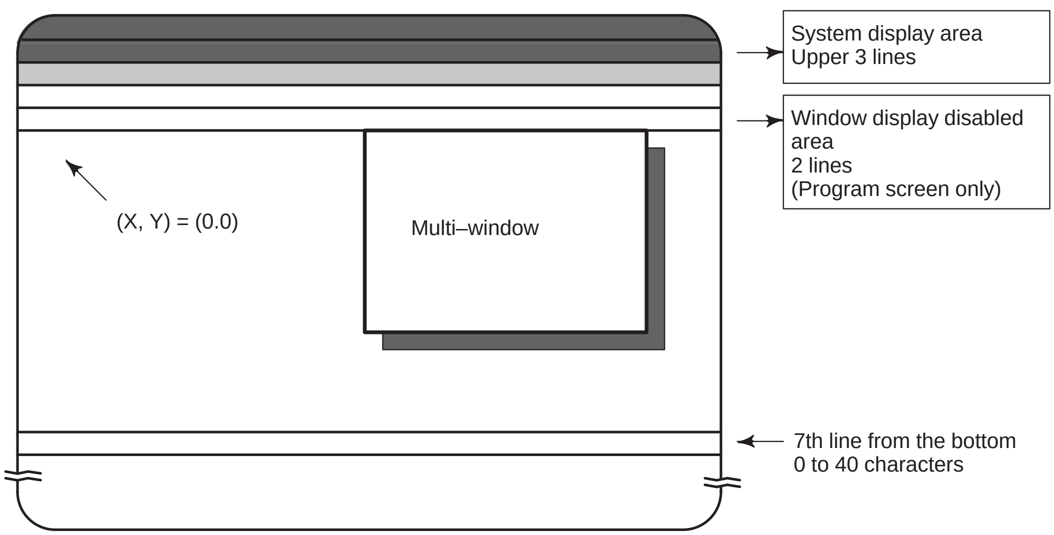
The basic menu screen, which is standard for conversation, is not displayed. The program coded in O1020 (O1021) is displayed.

Multi–window and basic menu screen display areas

With Super CAP II T, the multi–window cannot be displayed in the system display area (upper three lines) and the two lines below the system display area. Moreover, a basic menu screen created by each machine tool builder cannot be displayed in the system display area (upper three lines).

In connection with the above, set the macro executor compile parameter (bit 1 (LM30) of parameter No. 9009) to 0 (so that the full 30–line screen is not used for VGA display).

The method of screen address specification for character and graphics display is compatible with the conventional method for Super CAP T, so that a program can be transported without making major modifications to the program.



## E.12.3 Modifications Made to the Link File

To reflect the conversational system capacity and macro capacity expansions and to ensure compatibility with the VGA graphics capability, the link file has been partially changed.

```
/*
/* Conversation MACRO Sample LINK control file.
/*
/*
/* executor file 'FS16TC for VGA'
/*
CNC =c:\mcomp\BH18_Z0.EX1
CNC2 =c:\mcomp\BH18_Z0.EX2
CNC3 =c:\mcomp\BH18_Z0.EX3
PCODE=REAR
/*
/* compile parameter P9000 – P9009 and P9010 – P9059
/*
P9000=11000000
P9001=10000001
P9002=11000000
P9003=10001001
P9007=01000000
P9009=00000100
/*
P9013=200
/*
P9023=320
P9024=321
/*
P9030=27
P9031=28
P9032=29
/*
P9033=97
P9031=28
P9032=29
/*
P9033=97
P9037=7
P9044=2044
/*
P9038=4999
/*
/* Link file's
/*
FILE=CPS2T_Z0
FILE=BH18_Z0
FILE=TRANS
```

PCODE=REAR: Must not be omitted.

P9000: Set 3M (by setting bit 6 to 1).

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 9000 | LD6 | R3M | R2M | R1M | R512 | R256 | R128 | VRFY |

R3M = 1: Output to a 3M–byte ROM cassette or ROM module.

P9009: Number of screen lines used
The higher of these two bits must always be set to 0.
The lower of these two bits can be set arbitrarily by the machine tool builder.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 9009 | | | | | | CM30 | LM30 | |

LM30 = 1: The full 30–line screen is used for VGA display. (Super CAP II T)
LM30 = 0: The full 30–line screen is not used for VGA display. (Super CAP II T)
CM30 = 1: The full 30–line screen is used for VGA display. (CUSTOM)
CM30 = 0: The full 30–line screen is not used for VGA display. (CUSTOM)

**NOTE**
The modifications made to the link file described above are just an example. Set those items other than those indicated by an arrow and comments, as required, as described in this manual.

# F MACRO VARIABLES USED IN THE MACRO EXECUTOR FUNCTIONS

| Variable No. | Function | R/W | Conversa-tional | Auxiliary | Execut-able |
|---|---|---|---|---|---|
| **Macro varlables** | | | | | |
| #1 - #33 | Local variables | R/W | × | × | ○ |
| #1 - #99 | Array-type variables | R/W | ○ | ○ | × |
| #100 - #149 | Common variables (non-hold type) | R/W | ○ | ○ | ○ |
| #500 - #531 | Common variables (hold type) | R/W | ○ | ○ | ○ |
| #10000 - | P-CODE variables | R/W | ○ | ○ | ○ |
| #20000 - | Extended P-CODE variables | R/W | ○ | ○ | ○ |
| **Reading the remaing traveling distance** | | | | | |
| #5181 to #5188 | Variables for reading the remaining traveling distance | R/ | ○ | ○ | × |
| **Execution control variables** | | | | | |
| #8500 | Variable 1 for controlling the execution of conversational macros (CUSTOM screen 1) | R/W | ○ | ○ | × |
| #8550 | Variable 2 for controlling the execution of conversational macros (CUSTOM screen 2) | R/W | ○ | ○ | × |
| #8551 | Variable 3 for controlling the execution of conversational macros (CUSTOM screen 3) | R/W | ○ | ○ | × |
| **Key/data input control** | | | | | |
| #8501 | Key input control variable | R/ | ○ | × | × |
| #8502 | Data input control variable | R/W | ○ | × | × |
| #8503 | Numeric data variable | R/ | ○ | × | × |
| #8504 | Address data variable | R/ | ○ | × | × |
| #8552 | Variable for controlling extended data input for conversational macros | R/W | ○ | × | × |
| **Cursor control** | | | | | |
| #8505 | Cursor control variable | R/W | ○ | ○ | × |
| #8506 | Cursor X position control variable | R/W | ○ | ○ | × |
| #8507 | Cursor Y position control variable | R/W | ○ | ○ | × |
| **Screen control** | | | | | |
| #8509 | Variable for controlling the character string cataloging program | R/W | ○ | ○ | × |
| #8510 | Variable for controlling conversational macro function screens | R/W | ○ | ○ | × |
| **Processing of P–CODE varlables of array type** | | | | | |
| #8511 | Transfer source data | R/W | ○ | ○ | × |
| #8512 | Two-dimensional array number (transfer source) | R/W | ○ | ○ | × |
| #8513 | Three-dimensional array number (transfer source) | R/W | ○ | ○ | × |
| #8514 | Two-dimensional array number (transfer destination) | R/W | ○ | ○ | × |
| #8515 | Three-dimensional array number (transfer destination) | R/W | ○ | ○ | × |
| #8516 | Maximum number of one-dimensional array elements | R/W | ○ | ○ | × |
| #8517 | Maximum number of two-dimensional array elements | R/W | ○ | ○ | × |

R : Readable W : Writable ○ : Usable × : Unusable

| Variable No. | Function | R/W | Conversa-tional | Auxiliary | Execut-able |
|---|---|---|---|---|---|
| #8519 | Number of the first variable in the array | R/W | ○ | ○ | × |
| **Reading and writing an NC program** | | | | | |
| #8520 | Program number specification | R/W | ○ | ○ | × |
| #8521 | Block number specification | R/W | ○ | ○ | × |
| #8522 | Stored variable number specification | R/W | ○ | ○ | × |
| #8523 | Variable for specifying the number of decimal places | R/W | ○ | ○ | × |
| #8529 | Completion code for reading or writing an NC program | R/ | ○ | ○ | × |
| **Reading data stored in the tape** | | | | | |
| #8526 | Variable for reading the background editing status | R/ | ○ | ○ | × |
| #8527 | Variable for reading the number of cataloged programs | R/ | ○ | ○ | × |
| #8528 | Variable for reading the size of the free space in the CNC program memory | R/ | ○ | ○ | × |
| **Reader/punch interface** | | | | | |
| #8539 | Completion code for reader/punch interface | R/ | ○ | ○ | × |
| **MDI key image read function** | | | | | |
| #8549 | MDI key image storing variable | R/ | ○ | × | × |
| **Reading and resetting the cutting time and cutting distance** | | | | | |
| #8553 | Reading and presetting the cutting time | R/W | ○ | ○ | × |
| #8554 | Reading and presetting the cutting distance | R/W | ○ | ○ | × |
| **Key–in line control** | | | | | |
| #8561 | X coordinate of the point where the key-in line is displayed | R/W | ○ | × | × |
| #8562 | Y coordinate of the point where the key-in line is displayed | R/W | ○ | × | × |
| #8563 | Number of input keys | R/W | ○ | × | × |
| #8564 | Display of the prompt | R/W | ○ | × | × |
| #8565 | Display of the key-in line | R/W | ○ | × | × |
| **Interlock control for a signal axis dlrection** | | | | | |
| #8600 | Interlock control variable for a single axis direction | R/W | ○ | ○ | × |
| #8601 | Variable indicating the axis and direction of movement when the SKIP signal is turned on | R/ | ○ | ○ | × |
| **PMC axis control** | | | | | |
| #8602 | PMC control axis selection variable (with G codes) | R/W | ○ | ○ | × |
| #8700 | PMC control axis selection variable (with macro variables) | R/W | ○ | ○ | × |
| #8710 | PMC command signal variable (Area A) | R/W | ○ | ○ | × |
| #8711 | PMC control command variable (Area A) | R/W | ○ | ○ | × |
| #8712 | PMC cutting feedrate variable (Area A) | R/W | ○ | ○ | × |
| #8713 | Variable for traveling distance controlled by PMC (Area A) | R/W | ○ | ○ | × |
| #8715 | Read variable for PMC status signal (Area A) | R/ | ○ | ○ | × |
| #8720 | PMC command signal variable (Area B) | R/W | ○ | ○ | × |
| #8721 | PMC control command variable (Area B) | R/W | ○ | ○ | × |
| #8722 | PMC cutting feedrate variable (Area B) | R/W | ○ | ○ | × |
| #8723 | Variable for traveling distance controlled by PMC (Area B) | R/W | ○ | ○ | × |
| #8725 | Read variable for PMC status signal (Area B) | R/ | ○ | ○ | × |
| #8730 | PMC command signal variable (Area C) | R/W | ○ | ○ | × |
| #8731 | PMC control command variable (Area C) | R/W | ○ | ○ | × |

R : Readable W : Writable ○ : Usable × : Unusable

| Variable No. | Function | R/W | Conversa-tional | Auxiliary | Execut-able |
|---|---|---|---|---|---|
| #8732 | PMC cutting feedrate variable (Area C) | R/W | ○ | ○ | × |
| #8733 | Variable for traveling distance controlled by PMC (Area C) | R/W | ○ | ○ | × |
| #8735 | Read variable for PMC status signal (Area C) | R/ | ○ | ○ | × |
| #8740 | PMC command signal variable (Area D) | R/W | ○ | ○ | × |
| #8741 | PMC control command variable (Area D) | R/W | ○ | ○ | × |
| #8742 | PMC cutting feedrate variable (Area D) | R/W | ○ | ○ | × |
| #8743 | Variable for traveling distance controlled by PMC (Area D) | R/W | ○ | ○ | × |
| #8745 | Read variable for PMC status signal (Area D) | R/ | ○ | ○ | × |
| **Torque limit control** | | | | | |
| #8621 | Override value of the torque limit for the 1st servo axis | R/W | ○ | ○ | ○ |
| #8622 | Override value of the torque limit for the 2nd servo axis | R/W | ○ | ○ | ○ |
| #8623 | Override value of the torque limit for the 3rd servo axis | R/W | ○ | ○ | ○ |
| #8624 | Override value of the torque limit for the 4th servo axis | R/W | ○ | ○ | ○ |
| #8625 | Override value of the torque limit for the 5th servo axis | R/W | ○ | ○ | ○ |
| #8626 | Override value of the torque limit for the 6th servo axis | R/W | ○ | ○ | ○ |
| #8627 | Override value of the torque limit for the 7th servo axis | R/W | ○ | ○ | ○ |
| #8628 | Override value of the torque limit for the 8th servo axis | R/W | ○ | ○ | ○ |
| **Reading A/D converter data (Series 16/18)** | | | | | |
| #8631 | A/D converter data for channel 1 | R/W | ○ | ○ | × |
| #8632 | A/D converter data for channel 2 | R/W | ○ | ○ | × |
| #8633 | A/D converter data for channel 3 | R/W | ○ | ○ | × |
| #8634 | A/D converter data for channel 4 | R/W | ○ | ○ | × |
| **Window function** | | | | | |
| #8998 | System information ID | R/W | ○ | ○ | × |
| #8999 | System information | R/ | ○ | ○ | × |
| **Reading custom macro variables** | | | | | |
| #99000 – #99999 | #99000 + Custom macro variable number | R/W | ○ | ○ | ○ |
| **Offset memory and wark piece coordinate system (Series 16/18)** | | | | | |
| #100000 – | Extended system variable for the offset memory and workpiece coordinate system | R/W | ○ | ○ | ○ |

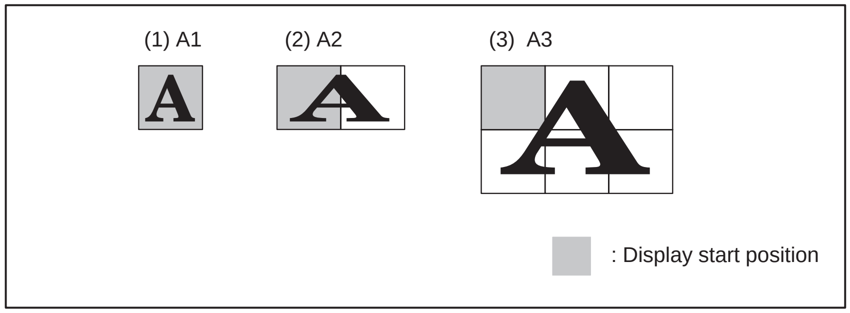
R : Readable W : Writable ○ : Usable × : Unusable

# G DIFFERENCES BETWEEN Series 0 AND Series 16/18 (CONVERSION FROM FS0 TO FS16/18)

## G.1 CHARACTER DISPLAY (G243)

Double size can be specified for character size specification A (only for alphanumeric characters).

⇒ An alphanumeric character is displayed having the same size as a kanji character.




## G.2 DISPLAY COLOR SPECIFICATION (G240)

(1) Setting a negative value for the display color reverses the display.

G240 Pp;

P : 0=Black
1=Red 2=Green 3=Yellow 4=Blue 5=Purple 6=Blue-green 7=White
1=Red 2=Green 3=Yellow 4=Blue 5=Purple 6=Blue-green 7=White

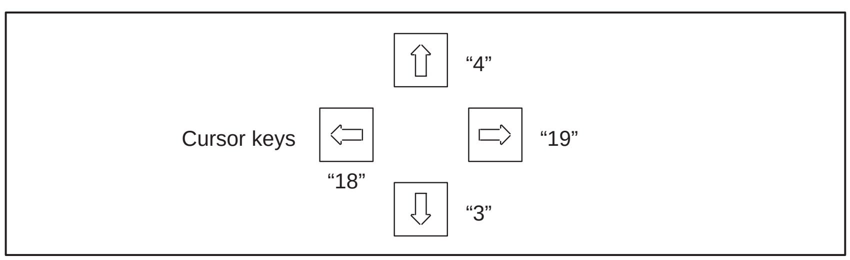
(2) Blinking display can be specified in address L.

G240 Pp Ll; L : 0 = Does not blink 1 = Blinks

⇒ The display blinks under the control of the hardware. The program does not have to contain a loop. Once blinking is specified for the display, it keeps blinking.

## G.3 KEY INPUT VARIABLE (#8501)

(1) Key numbers are added for the cursor keys.




(2) Key number 9 cannot be used because there is not a START key on the MDI.

## G.4 CONTROLLING CONVERSATIONAL MACRO FUNCTION SCREENS (#8510)

Screen numbers and corresponding screens are changed.

| For standard MDI | | For small MDI | |
|---|---|---|---|
| 0 | POS | 0 | POS |
| 1 | PROG | 1 | PROG |
| 2 | OFFSET/SETTING | 2 | OFFSET/SETTING |
| 3 | SYSTEM | 3 | SYSTEM |
| 4 | MESSAGE | 4 | MESSAGE |
| 5 | GRAPHIC | 5 | CUSTOM/GRAPHIC |
| 6 | CUSTOM | 6 | CUSTOM |
| 7 | FAPT | | |

⇒When the graphic screen is provided, the graphic screen is selected.)

## G.5 PMC WRITE CONTROL CODE

Signed values consisting of 1 to 4 bytes can be written.

G310 Dd Lℓ Qq ;
G310 Rr Lℓ Qq ;
G310 Cc Lℓ Qq ;
G310 Kk Lℓ Qq ;

D : Data table number of the PMC
R : Internal relay number of the PMC
C : Counter
K : Keep relay
L : Data length (1 or blank: 1 byte, 2: 2 bytes, 4: 4 bytes)
⇒ Unsigned when L is not specified. Signed when L is 1.
Q : Write data (Converted to binary when transferred)

## G.6 PMC READ CONTROL CODE

Signed values consisting of 1 to 4 bytes can be read.

G310 Dd Pp Lℓ ;
G310 Rr Pp Lℓ ;
G310 Cc Pp Lℓ ;
G310 Kk Pp Lℓ ;

D : Data table number of the PMC
R : Internal relay number of the PMC
C : Counter
K : Keep relay
P : Number of the variable in which the read data is stored.
L : Data length (1 or blank: 1 byte, 2: 2 bytes, 4: 4 bytes)
⇒ Unsigned when L is not specified. Signed when L is 1.
⇒ The value is handled using two's complement.

## G.7 CALLING A SUBPROGRAM WITH AN AXIS ADDRESS

| Compile parameter | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 9005 | TMACC | | | AXCLS | AX4CL | AX3CL | AX2CL | AX1CL |

AX1CL 0 : Subprogram call by the 1st address is invalid
1 : Subprogram call by the 1st address is valid

AX2CL 0 : Subprogram call by the 2nd address is invalid
1 : Subprogram call by the 2nd address is valid

AX3CL 0 : Subprogram call by the 3rd address is invalid
1 : Subprogram call by the 3rd address is valid

AX4CL 0 : Subprogram call by the 4th address is invalid
1 : Subprogram call by the 4th address is valid

AXCLS 0 : Always program O9009 is called irrespective of specified axes.
1 : The program number to be called depends on a specified axis:
Program O9031 is called when 1st axis is specified.
Program O9032 is called when 2nd axis is specified.
: :
Program O9038 is called when 8th axis is specified.

TMACC 0 : Macro call by T code is invalid
1 : Macro call by T code is valid

| Compile parameter | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 9008 | HRGCC | | | | AX8CL | AX7CL | AX6CL | AX5CL |

AX5CL 0 : Subprogram call by the 5th axis address is made invalid.
1 : Subprogram call by the 5th axis address is made valid.

AX6CL 0 : Subprogram call by the 6th axis address is made invalid.
1 : Subprogram call by the 6th axis address is made valid.

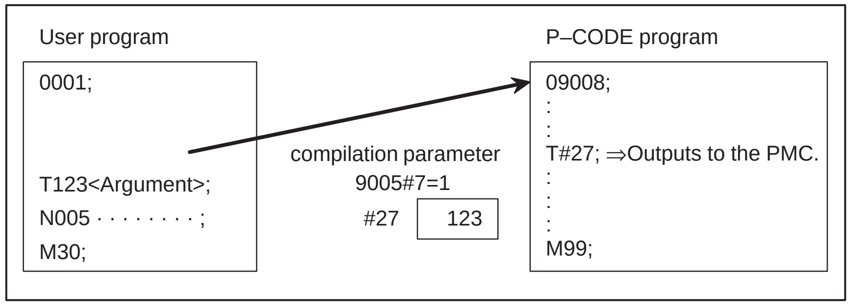
AX7CL 0 : Subprogram call by the 7th axis address is made invalid.
1 : Subprogram call by the 7th axis address is made valid.

AX8CL 0 : Subprogram call by the 8th axis address is made invalid.
1 : Subprogram call by the 8th axis address is made valid.

HRGCC 0 : Character display screen is not set to the intensity modulation mode.
1 : Character display screen is set to the intensity modulation mode.

## G.8 CALLING A MACRO WITH A T CODE

P-CODE program O9008 can be called from the user program using a T code.




| Compilation parameter | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 9007 | TMAC | | | | | | | |

#7(TMAC) 0 : Does not call a macro with a T code.
1 : Calls a macro with a T code.

⇒ The specified T code is assigned to #27.

⇒ Addresses P (#16) and L (#12) can also be used as arguments.

⇒ Up to five G codes (including one code for a group) are assigned to #28 to #32.

| | |
|---|---|
| General argument | #1 to #26 |
| T code | #27 |
| G code | #28 to #32 |

(Example)

G91 G28 X123.45678 T5678;⇒

Local variable

| | |
|---|---|
| #24 | 123.45678 |
| #27 | 5678 |
| #28 | 28 |
| #29 | 91 |

## G.9 A BIT CANNOT BE SPECIFIED TO READ A PARAMETER.

In FS0, a command such as #100=P100.2; can be specified. In FS16, however, use a command such as #100=[P100 AND 4]/4; instead.

## G.10 THE ROM SIZE IS SPECIFIED DIFFERENTLY.

| Compile parameter | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 9000 | | | | R1MB | R512 | R256 | | |

#4 (R1MB) Writes data in a 1M-byte ROM module.

#3 (R512) Writes data in a 512K-byte ROM module.

#2 (R256) Writes data in a 256K-byte ROM module.

## G.11 THE USE OF P–CODE DEDICATED VARIABLES HAS BEEN EXTENDED.

Compile parameter

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 9002 | | | | | EVF | | | |

#3 (EVF) 0 : Extended P-CODE variables (#20000 or larger) are floating-point type.

1 : Extended P-CODE variables (#20000 or larger) are integer type.
(Numeric values –32768 to +32767 can be handled.)

**NOTE**
When extended P-CODE variables are integer type, note the following:
1 When an extended P-CODE variable is specified on the left side of an expression, the result of calculation is rounded off and assigned.
2 When an extended P-CODE variable is referenced in an expression, it is converted to floating-point type.

Compile parameter

| | |
|---|---|
| 9037 | Number of P–CODE dedicated variables to be used (0 to 60) |

Sets the number of P-CODE dedicated variables to be used in units of 100. Each 100 variables use 1.63m of tape storage.

9037 = 12 or less for 20m tape storage

9037 = 25 or less for 40m tape storage

9037 = 51 or less for 80m tape storage

9037 = 60 or less for 160m to 1280m tape storage

Compile parameter

| | |
|---|---|
| 9044 | Number of extended P–CODE dedicated variables to be used |

Sets the number of extended P-CODE dedicated variables to be used. Twelve floating-point variables or 30 integer variables make a set. Each set uses 0.21m of tape storage.

9044 = 819 or less for 160m tape storage

9044 = 1638 or less for 320m tape storage

9044 = 3276 or less for 640m tape storage (9002#3 = 0)

9044 = 2184 or less for 640m tape storage (9002#3 = 1)

9044 = 5461 or less for 1280m tape storage (9002#3 = 0)

9044 = 2184 or less for 1280m tape storage (9002#3 = 1)

## G.12 THE METHOD FOR DISPLAYING P–CODE VARIABLES #10000 OR LARGER HAS BEEN CHANGED.

| Executor parameter | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 9000 | | | | | | | NDP | |

#1 (NDP) 0 : Local and common variables for a P-CODE program are not displayed.

1 : Local and common variables for a P-CODE program are displayed.
(Press the OFFSET key several times.)

```
P–CODE VARIABLE

    No.      DATA               No.      DATA
    100       0.000             110       0.000
    101       9.000             111       9.000
    102      10.000             112      10.000
    103                         113
    104                         114
    105                         115
    106                         116
    107                         117
    108                         118
    109                         119

    NUM.    ________
  [      ] [      ] [      ] [      ] [NO–SEL]
```

Enter the variable number and press soft key [NO-SEL] to select the number of the variable to be displayed.

⇒ The values of the variables just before the screen is displayed are displayed. Note that even if the value of a variable is changed after the screen is displayed, the displayed value does not change.

⇒ In FS16, variables of numbers #10000 or larger can be displayed using soft key [NO-SEL]. To display P-CODE dedicated variables of numbers #10000 or larger in FS0, set the number of the first variable to be displayed in parameter 9001. Twenty variables after and including the set variable are displayed.

# H G CODES THAT CAN BE USED BY THE CONVERSATIONAL AND AUXILIARY MACROS

**NOTE**

1 The conversational macro can issue all G codes. The auxiliary macro cannot issue commands relating to screen display (●). The execution macro cannot issue G codes. (G01, G02, and G03 are commands for linear interpolation and cylindrical interpolation of the NC unit.)

2 A single-shot G code is marked with "1S". A continuous-state G code is marked with "M". Each continuous-state G code is shared by the conversational and auxiliary macros and belongs to one G-code group.

| G code | Function | Standard command format | M/1S | Conversa-tional | Auxiliary |
|---|---|---|---|---|---|
| G01 | Graphics: Displays a straight line. | G01 X_ Y_ ; | M | ○ | ● |
| G02 | Graphics: Displays an arc (clockwise). | G02 X_ Y_ I_ J_ Q_ ; | M | ○ | ● |
| G03 | Graphics: Displays an arc (counterclockwise). | G03 X_ Y_ I_ J_ Q_ ; | M | ○ | ● |
| | | | | | |
| G202 | Erases the screen. | G206 P_ ; | 1S | ○ | ● |
| G206 | Graphics: Fills in an area. | G206 P_ X_ Y_ ; | 1S | ○ | ● |
| G240 | Screen/graphics: Specifies a display color. | G240 P_ L_ ; | 1S | ○ | ● |
| G242 | Graphics: Specifies a start point. | G242 X_ Y_ ; | M | ○ | ● |
| G243 | Displays characters. | G243 [String Form] ; | M | ○ | ● |
| G244 | Graphics: Specifies a line type. | G244 P_ ; | 1S | ○ | ● |
| | | | | | |
| G310 | Reads or writes the PMC data. | G310 [R/D/C/K]_ [Q/P]_ L_ ; | M | ○ | ○ |
| G315 | Processes the array-type data. | G315 P_ K_ ; | 1S | ○ | ○ |
| G319 | Registration of external character | G319 P_ Q_ ; | 1S | ○ | ○ |
| G320 | Reference to the NC program: Catalogs a program. | G320 ; | 1S | ○ | ○ |
| G321 | Reference to the NC program: Deletes a program. | G321 ; | 1S | ○ | ○ |
| G325 | Reference to the NC program: Reads a block. | G325 ; | 1S | ○ | ○ |
| G326 | Reference to the NC program: Writes a block. | G326 P_ ; | 1S | ○ | ○ |
| G327 | Reference to the NC program: Deletes a block. | G327 ; | 1S | ○ | ○ |
| G328 | Reference to the NC program: Reads a block (characters). | G328 ; | 1S | ○ | ○ |
| G329 | Reference to the NC program: Writes a block (characters). | G329 P_ ; | 1S | ○ | ○ |

| G code | Function | Standard command format | M/1S | Conversa-tional | Auxiliary |
|---|---|---|---|---|---|
| G330 | RS232c: Opens a line. | G330 P_ B_ S_ C_ (F_/L_) ; | 1S | ○ | ○ |
| G331 | RS232c: Closes a line. | G331 ; | 1S | ○ | ○ |
| G335 | RS232c: Reads a single character. (Reception) | G335 P_ ; | 1S | ○ | ○ |
| G336 | RS232c: Writes data. (Transmission) | G336 [String Form] ; | M | ○ | ○ |
| G337 | RS232c: Reads variable data. (Reception) | G337 P_ Q_ R_ ; | 1S | ○ | ○ |
| G338 | RS232c: Writes variable data. (Transmission) | G338 P_ Q_ F_ Z_ R_ ; | 1S | ○ | ○ |
| G339 | RS232c: Controls the FANUC cassette. | G339 P_ F_ L_ S_ ; | 1S | ○ | ○ |
| | | | | | |
| G340 | PMC axis control: Issues a rapid traverse command. | G340 X_ ; | 1S | ○ | ○ |
| G341 | PMC axis control: Issues a cutting feed command. | G341 X_ F_ ; | 1S | ○ | ○ |
| G344 | PMC axis control: Issues a dwell command. | G344 X_ ; | 1S | ○ | ○ |
| G345 | PMC axis control: Issues a reference position return command. | G345 ; | 1S | ○ | ○ |
| G346 | PMC axis control: Issues a miscellaneous function command. | G346 M_ ; | 1S | ○ | ○ |
| G348 | PMC axis control: Issues a status signal read command. | G348 P_ ; | 1S | ○ | ○ |
| G349 | PMC axis control: Issues a command signal write command. | G349 P_ ; | 1S | ○ | ○ |

**NOTE**

1 G codes (G340 to G349) for PMC axis control cannot be used for Series 20-TA.

2 External character registration (G319) cannot be used for Series 16/18.

3 G codes for graphic display (G01, G02, G03, G204, G206, G242, G244, G249, etc.) cannot be used with the Series 21 or 20–MA.

# I INTERNAL CODE

| | | | | |
|---|---|---|---|---|
| 0020 | 0021 | 0022 | 0023 | 0024 |
| | ! | ” | # | $ |
| 0025 | 0026 | 0027 | 0028 | 0029 |
| % | & | ’ | ( | ) |
| 002A | 002B | 002C | 002D | 002E |
| * | + | , | - | . |
| 002F | 0030 | 0031 | 0032 | 0033 |
| / | 0 | 1 | 2 | 3 |
| 0034 | 0035 | 0036 | 0037 | 0038 |
| 4 | 5 | 6 | 7 | 8 |
| 0039 | 003A | 003B | 003C | 003D |
| 9 | : | ; | < | = |
| 003E | 003F | 0040 | 0041 | 0042 |
| > | ? | @ | A | B |
| 0043 | 0044 | 0045 | 0046 | 0047 |
| C | D | E | F | G |
| 0048 | 0049 | 004A | 004B | 004C |
| H | I | J | K | L |
| 004D | 004E | 004F | 0050 | 0051 |
| M | N | O | P | Q |
| 0052 | 0053 | 0054 | 0055 | 0056 |
| R | S | T | U | V |
| 0057 | 0058 | 0059 | 005A | 005B |
| W | X | Y | Z | [ |
| 005C | 005D | 005E | 005F | 0061 |
| ¥ | ] | ^ | _ | a |
| 0062 | 0063 | 0064 | 0065 | 0066 |
| b | c | d | e | f |
| 0067 | 0068 | 0069 | 006A | 006B |
| g | h | i | j | k |
| 006C | 006D | 006E | 006F | 0070 |
| l | m | n | o | p |
| 0071 | 0072 | 0073 | 0074 | 0076 |
| q | r | s | t | u |

| | | | | |
|---|---|---|---|---|
| 0076 | 0077 | 0078 | 0079 | 007A |
| v | w | x | y | z |
| 00A0 | 00A1 | 00A2 | 00A3 | 00A4 |
| ～ | 。 | 「 | 」 | 、 |
| 00A5 | 00A6 | 00A7 | 00A8 | 00A9 |
| ・ | ヲ | ァ | ィ | ゥ |
| 00AA | 00AB | 00AC | 00AD | 00AE |
| ェ | ォ | ャ | ュ | ョ |
| 00AF | 00B0 | 00B1 | 00B2 | 00B3 |
| ッ | ー | ア | イ | ウ |
| 00B4 | 00B5 | 00B6 | 00B7 | 00B8 |
| エ | オ | カ | キ | ク |
| 00B9 | 00BA | 00BB | 00BC | 00BD |
| ケ | コ | サ | シ | ス |
| 00BE | 00BF | 00C0 | 00C1 | 00C2 |
| セ | ソ | タ | チ | ツ |
| 00C3 | 00C4 | 00C5 | 00C6 | 00C7 |
| テ | ト | ナ | ニ | ヌ |
| 00C8 | 00C9 | 00CA | 00CB | 00CC |
| ネ | ノ | ハ | ヒ | フ |
| 00CD | 00CE | 00CF | 00D0 | 00D1 |
| ヘ | ホ | マ | ミ | ム |
| 00D2 | 00D3 | 00D4 | 00D5 | 00D6 |
| メ | モ | ヤ | ユ | ヨ |
| 00D7 | 00D8 | 00D9 | 00DA | 00DB |
| ラ | リ | ル | レ | ロ |
| 00DC | 00DD | 00DE | 00DF | 2137 |
| ワ | ン | ゛ | ゜ | 〃 |
| 2421 | 2422 | 2423 | 2424 | 2425 |
| ぁ | あ | ぃ | い | ぅ |
| 2426 | 2427 | 2428 | 2429 | 242A |
| う | ぇ | え | ぉ | お |
| 242B | 242C | 242D | 242E | 242F |
| か | が | き | ぎ | く |

| | | | | |
|---|---|---|---|---|
| 2430 | 2431 | 2432 | 2433 | 2434 |
| ぐ | け | げ | こ | ご |
| 2435 | 2436 | 2437 | 2438 | 2439 |
| さ | ざ | し | じ | す |
| 243A | 243B | 243C | 243D | 243E |
| ず | せ | ぜ | そ | ぞ |
| 243F | 2440 | 2441 | 2442 | 2443 |
| た | だ | ち | ぢ | っ |
| 2444 | 2445 | 2446 | 2447 | 2448 |
| つ | づ | て | で | と |
| 2449 | 244A | 244B | 244C | 244D |
| ど | な | に | ぬ | ね |
| 244E | 244F | 2450 | 2451 | 2452 |
| の | は | ば | ぱ | ひ |
| 2453 | 2454 | 2455 | 2456 | 2457 |
| び | ぴ | ふ | ぶ | ぷ |
| 2458 | 2459 | 245A | 245B | 245C |
| へ | べ | ぺ | ほ | ぼ |
| 245D | 245E | 245F | 2460 | 2461 |
| ぽ | ま | み | む | め |
| 2462 | 2463 | 2464 | 2465 | 2466 |
| も | ゃ | や | ゅ | ゆ |
| 2467 | 2468 | 2469 | 246A | 246B |
| ょ | よ | ら | り | る |
| 246C | 246D | 246E | 246F | 2472 |
| れ | ろ | ゎ | わ | を |
| 2473 | 2641 | 2642 | 2F40 | 2F41 |
| ん | α | β | → | ↗ |
| 2F42 | 2F43 | 2F44 | 2F45 | 2F46 |
| ↑ | ↖ | ← | ↙ | ↓ |
| 2F47 | 2F48 | 2F49 | 2F4A | 2F4B |
| ↘ | ↻ | ↺ | ⌒ | ◯ |
| 2F4C | 2F50 | 2F51 | 2F52 | 2F53 |
| ■ | ▽ | ▽▽ | ▽▽▽ | ▽▽▽▽ |
| 2F60 | 2F61 | 2F62 | 2F63 | 2F64 |
| ─○─ | ┬ | ┐ | ┤ | ┘ |
| 2F65 | 2F66 | 2F67 | 2F68 | 2F69 |
| ┴ | └ | ├ | ┌ | ￣ |
| 2F6A | 2F6B | 2F6C | 2F6D | 2F6E |
| │ | ┼ | ┤ | ├ | ⫽ |

| | | | | |
|---|---|---|---|---|
| 2F6F | 2F70 | 2F71 | 2F72 | 2F73 |
| ⫽ | ( | ) | [ | ] |
| 2F74 | 2F75 | 2F76 | 2F77 | 2F78 |
| ┐ | ┌ | ─ | ─ | ↑ |
| 2F79 | 2F7A | 2F7B | | |
| ↑ | ↑ | _ | | |
| | 302E | 3035 | 3037 | 3042 |
| | 握 | 圧 | 扱 | 安 |
| 3045 | 304A | 304C | 304F | 3055 |
| 暗 | 以 | 位 | 囲 | 意 |
| 305B | 305C | 3063 | 3068 | 306C |
| 異 | 移 | 違 | 域 | 一 |
| 3075 | 307A | 3122 | 3123 | 3126 |
| 印 | 引 | 陰 | 隠 | 右 |
| 313F | 3146 | 3154 | 315B | 315F |
| 運 | 影 | 鋭 | 越 | 円 |
| 3168 | 316F | 3173 | 317A | 317B |
| 沿 | 縁 | 遠 | 凹 | 央 |
| 317C | 317D | 317E | 3221 | 3223 |
| 奥 | 往 | 応 | 押 | 横 |
| 323D | 322F | 3230 | 3239 | 323C |
| 化 | 億 | 屋 | 温 | 下 |
| 323D | 323E | 323F | 3241 | 3243 |
| 化 | 仮 | 何 | 価 | 加 |
| 3244 | 3248 | 324A | 324C | 3254 |
| 可 | 家 | 科 | 果 | 稼 |
| 3255 | 3259 | 325D | 3261 | 3268 |
| 箇 | 荷 | 課 | 過 | 画 |
| 3271 | 3272 | 3273 | 327E | 3323 |
| 会 | 解 | 回 | 改 | 械 |
| 3326 | 3328 | 332B | 332C | 3330 |
| 界 | 絵 | 開 | 階 | 外 |
| 3335 | 333A | 3346 | 3348 | 334A |
| 概 | 該 | 各 | 拡 | 格 |
| 334B | 334E | 3351 | 3354 | 3356 |
| 核 | 確 | 角 | 郭 | 隔 |
| 3358 | 335B | 335D | 3364 | 3368 |
| 学 | 額 | 掛 | 割 | 活 |
| 342C | 3430 | 3439 | 3441 | 3446 |
| 巻 | 完 | 換 | 漢 | 監 |

| | | | | |
|---|---|---|---|---|
| 3449 | 344A | 3451 | 3453 | 3456 |
| 管 | 簡 | 観 | 貫 | 間 |
| 3458 | 345D | 345E | 346A | 346F |
| 関 | 丸 | 含 | 願 | 器 |
| 3470 | 3471 | 3473 | 3474 | 347B |
| 基 | 奇 | 寄 | 岐 | 既 |
| 347C | 3521 | 3522 | 3524 | 352C |
| 期 | 機 | 帰 | 気 | 規 |
| 352D | 352F | 3530 | 353B | 353F |
| 記 | 起 | 軌 | 技 | 疑 |
| 3541 | 3551 | 3552 | 3555 | 3559 |
| 義 | 却 | 客 | 逆 | 休 |
| 355E | 3561 | 3565 | 3566 | 3569 |
| 急 | 求 | 球 | 究 | 級 |
| 356B | 356C | 356E | 3576 | 3577 |
| 給 | 旧 | 去 | 許 | 距 |
| 3621 | 3626 | 362D | 362F | 3635 |
| 供 | 共 | 境 | 強 | 教 |
| 3648 | 364A | 3651 | 3652 | 3658 |
| 業 | 曲 | 均 | 巾 | 禁 |
| 365A | 3661 | 3662 | 3668 | 366B |
| 筋 | 近 | 金 | 区 | 矩 |
| 366E | 3671 | 3675 | 3676 | 3679 |
| 駆 | 具 | 空 | 偶 | 隅 |
| 3721 | 372B | 3732 | 3738 | 3739 |
| 掘 | 繰 | 群 | 係 | 傾 |
| 373F | 3741 | 3742 | 374F | 3750 |
| 型 | 形 | 径 | 系 | 経 |
| 3751 | 3757 | 375A | 3765 | 3767 |
| 継 | 計 | 軽 | 桁 | 欠 |
| 3768 | 376A | 376B | 376F | 3821 |
| 決 | 穴 | 結 | 件 | 検 |
| 3822 | 3826 | 382A | 382B | 3833 |
| 権 | 研 | 肩 | 見 | 験 |
| 3835 | 3836 | 3839 | 383A | 383B |
| 元 | 原 | 弦 | 減 | 源 |
| 383D | 3840 | 3842 | 3844 | 3846 |
| 現 | 言 | 限 | 個 | 呼 |
| 3847 | 384A | 384C | 385F | 3865 |
| 固 | 己 | 弧 | 互 | 後 |

| | | | | |
|---|---|---|---|---|
| 3866 | 386C | 386D | 386E | 3872 |
| 御 | 語 | 誤 | 護 | 交 |
| 387A | 387C | 387D | 387E | 3926 |
| 効 | 厚 | 口 | 向 | 孔 |
| 3929 | 392A | 392D | 3933 | 3935 |
| 工 | 巧 | 広 | 抗 | 控 |
| 3939 | 393B | 393D | 3942 | 3945 |
| 更 | 校 | 構 | 溝 | 硬 |
| 3953 | 3954 | 395D | 395F | 3960 |
| 荒 | 行 | 鋼 | 降 | 項 |
| 3962 | 3966 | 3967 | 396F | 3970 |
| 高 | 号 | 合 | 刻 | 告 |
| 3975 | 397E | 3A2C | 3A2E | 3A38 |
| 黒 | 込 | 根 | 混 | 左 |
| 3A39 | 3A3F | 3A42 | 3A46 | 3A47 |
| 差 | 鎖 | 座 | 再 | 最 |
| 3A4E | 3A51 | 3A59 | 3A5F | 3A60 |
| 採 | 済 | 細 | 在 | 材 |
| 3A62 | 3A6E | 3A6F | 3A76 | 3B28 |
| 財 | 作 | 削 | 策 | 雑 |
| 3B32 | 3B33 | 3B36 | 3B3A | 3B3B |
| 参 | 山 | 散 | 産 | 算 |
| 3B44 | 3B45 | 3B48 | 3B4D | 3B4F |
| 残 | 仕 | 使 | 四 | 始 |
| 3B51 | 3B52 | 3B57 | 3B58 | 3B5F |
| 姿 | 子 | 思 | 指 | 止 |
| 3B65 | 3B67 | 3B69 | 3B6B | 3B6E |
| 糸 | 紫 | 脂 | 視 | 試 |
| 3B71 | 3B75 | 3B76 | 3B77 | 3B7D |
| 資 | 歯 | 事 | 似 | 持 |
| 3B7E | 3C21 | 3C23 | 3C28 | 3C2A |
| 時 | 次 | 治 | 示 | 耳 |
| 3C2B | 3C30 | 3C34 | 3C3A | 3C3C |
| 自 | 式 | 軸 | 失 | 室 |
| 3C41 | 3C42 | 3C4C | 3C4D | 3C50 |
| 質 | 実 | 写 | 射 | 斜 |
| 3C54 | 3C56 | 3C5A | 3C61 | 3C65 |
| 者 | 車 | 借 | 釈 | 弱 |
| 3C67 | 3C68 | 3C69 | 3C6A | 3C6C |
| 主 | 取 | 守 | 手 | 殊 |

| | | | | |
|---|---|---|---|---|
| 3C6F | 3C75 | 3C77 | 3C79 | 3C7D |
| 種 | 受 | 寿 | 樹 | 収 |
| 3C7E | 3D24 | 3D2A | 3D38 | 3D3D |
| 周 | 修 | 終 | 集 | 十 |
| 3D3E | 3D44 | 3D45 | 3D50 | 3D51 |
| 従 | 縦 | 重 | 出 | 術 |
| 3D52 | 3D60 | 3D63 | 3D67 | 3D68 |
| 述 | 準 | 純 | 順 | 処 |
| 3D69 | 3D6A | 3D71 | 3D75 | 3D78 |
| 初 | 所 | 書 | 助 | 序 |
| 3D7C | 3E21 | 3E26 | 3E2E | 3E2F |
| 除 | 勝 | 商 | 小 | 少 |
| 3E30 | 3E3A | 3E43 | 3E44 | 3E48 |
| 尚 | 昇 | 消 | 渉 | 照 |
| 3E4A | 3E4E | 3E4F | 3E5A | 3E5C |
| 省 | 称 | 章 | 証 | 詳 |
| 3E5D | 3E65 | 3E6A | 3E6C | 3E6F |
| 象 | 上 | 剰 | 場 | 常 |
| 3E72 | 3E75 | 3E7E | 3F22 | 3F27 |
| 条 | 状 | 飾 | 植 | 色 |
| 3F28 | 3F29 | 3F2D | 3F2E | 3F2F |
| 触 | 食 | 伸 | 信 | 侵 |
| 3F34 | 3F36 | 3F37 | 3F3B | 3F3C |
| 心 | 振 | 新 | 浸 | 深 |
| 3F3F | 3F47 | 3F48 | 3F4A | 3F4D |
| 真 | 診 | 身 | 進 | 人 |
| 3F4F | 3F5E | 3F62 | 3F64 | 3F65 |
| 刃 | 図 | 垂 | 推 | 水 |
| 3F6D | 3F74 | 3F78 | 4023 | 4029 |
| 錐 | 数 | 据 | 寸 | 制 |
| 402D | 402E | 4030 | 4035 | 4038 |
| 性 | 成 | 整 | 正 | 生 |
| 403A | 403D | 4044 | 404A | 4050 |
| 精 | 製 | 青 | 席 | 石 |
| 4051 | 4053 | 4056 | 405A | 405C |
| 積 | 績 | 赤 | 切 | 接 |
| 405E | 405F | 4061 | 4062 | 4064 |
| 折 | 設 | 節 | 説 | 絶 |
| 4068 | 4069 | 406C | 4075 | 407B |
| 先 | 千 | 専 | 浅 | 旋 |

| | | | | |
|---|---|---|---|---|
| 407E | 412A | 4130 | 4133 | 4134 |
| 線 | 選 | 前 | 然 | 全 |
| 4146 | 4147 | 414F | 4150 | 415B |
| 粗 | 素 | 創 | 双 | 想 |
| 415C | 415E | 4160 | 4161 | 416A |
| 捜 | 挿 | 操 | 早 | 相 |
| 416D | 4175 | 4176 | 4177 | 417D |
| 総 | 装 | 走 | 送 | 増 |
| 4226 | 4227 | 4228 | 422C | 422D |
| 側 | 則 | 即 | 測 | 足 |
| 422E | 4230 | 4233 | 4238 | 423B |
| 速 | 属 | 続 | 存 | 損 |
| 423E | 423F | 4240 | 4247 | 424E |
| 他 | 多 | 太 | 打 | 体 |
| 4250 | 4254 | 4256 | 4258 | 4260 |
| 対 | 待 | 態 | 替 | 退 |
| 4265 | 4266 | 4267 | 4268 | 426A |
| 代 | 台 | 大 | 第 | 題 |
| 426E | 4272 | 4323 | 432B | 4331 |
| 卓 | 択 | 達 | 谷 | 単 |
| 4335 | 433A | 433B | 433C | 4347 |
| 探 | 炭 | 短 | 端 | 断 |
| 434A | 434D | 434E | 434F | 4356 |
| 段 | 値 | 知 | 地 | 置 |
| 4357 | 4359 | 4365 | 4366 | 436C |
| 致 | 遅 | 着 | 中 | 柱 |
| 436D | 4372 | 437A | 4425 | 4427 |
| 注 | 鋳 | 丁 | 張 | 徴 |
| 4434 | 4436 | 4439 | 443A | 443B |
| 調 | 超 | 長 | 頂 | 鳥 |
| 443E | 4449 | 444C | 4463 | 4464 |
| 直 | 追 | 通 | 低 | 停 |
| 446A | 446C | 4478 | 4479 | 447B |
| 定 | 底 | 程 | 締 | 訂 |
| 452A | 452C | 4534 | 453A | 453E |
| 的 | 適 | 鉄 | 添 | 転 |
| 4540 | 4541 | 4545 | 4550 | 4553 |
| 点 | 伝 | 電 | 登 | 途 |
| 4559 | 456A | 4576 | 4579 | 4628 |
| 度 | 投 | 当 | 等 | 逃 |

| | | | | |
|---|---|---|---|---|
| 462C | 4630 | 4631 | 4633 | 463B |
| 頭 | 動 | 同 | 導 | 道 |
| 4640 | 4643 | 4648 | 4649 | 464C |
| 得 | 特 | 独 | 読 | 凸 |
| 464D | 465F | 4662 | 4679 | 467C |
| 突 | 鈍 | 内 | 肉 | 日 |
| 467E | 4724 | 4727 | 472E | 472F |
| 入 | 任 | 認 | 熱 | 年 |
| 473B | 473C | 473D | 474B | 474F |
| 濃 | 納 | 能 | 破 | 馬 |
| 4753 | 4754 | 4755 | 4758 | 475B |
| 排 | 敗 | 杯 | 背 | 配 |
| 475C | 4772 | 4776 | 4822 | 482F |
| 倍 | 白 | 薄 | 箱 | 発 |
| 4834 | 483C | 483D | 483E | 483F |
| 抜 | 伴 | 判 | 半 | 反 |
| 4842 | 4844 | 4846 | 484C | 484F |
| 搬 | 板 | 汎 | 般 | 範 |
| 4856 | 4866 | 4869 | 486F | 4873 |
| 番 | 比 | 皮 | 被 | 非 |
| 4877 | 4879 | 487E | 492C | 4934 |
| 備 | 微 | 美 | 必 | 百 |
| 4938 | 493D | 4941 | 4943 | 494A |
| 標 | 表 | 描 | 秒 | 品 |
| 4954 | 4955 | 495B | 4961 | 4969 |
| 不 | 付 | 布 | 普 | 負 |
| 4974 | 497A | 497B | 497C | 497D |
| 部 | 伏 | 副 | 復 | 幅 |
| 4A23 | 4A2A | 4A2C | 4A34 | 4A38 |
| 複 | 物 | 分 | 粉 | 文 |
| 4A39 | 4A3B | 4A3F | 4A42 | 4A44 |
| 聞 | 併 | 平 | 並 | 閉 |
| 4A47 | 4A4C | 4A51 | 4A52 | 4A54 |
| 頁 | 別 | 変 | 片 | 編 |
| 4A55 | 4A56 | 4A59 | 4A5B | 4A5D |
| 辺 | 返 | 勉 | 弁 | 保 |
| 4A62 | 4A64 | 4A6F | 4A71 | 4A73 |
| 歩 | 補 | 倣 | 包 | 報 |
| 4A7C | 4A7D | 4B21 | 4B3A | 4B40 |
| 放 | 方 | 法 | 忘 | 棒 |

| | | | | |
|---|---|---|---|---|
| 4B5C | 4B60 | 4B67 | 4B68 | 4B76 |
| 本 | 摩 | 枚 | 毎 | 末 |
| 4B7C | 4C24 | 4C29 | 4C35 | 4C3E |
| 万 | 未 | 密 | 無 | 名 |
| 4C3F | 4C40 | 4C47 | 4C4C | 4C57 |
| 命 | 明 | 滅 | 面 | 耗 |
| 4C5A | 4C5C | 4C61 | 4C64 | 4C67 |
| 木 | 目 | 戻 | 問 | 門 |
| 4C73 | 4C75 | 4C7D | 4D2D | 4D33 |
| 約 | 訳 | 油 | 有 | 由 |
| 4D3D | 4D3E | 4D3F | 4D46 | 4D49 |
| 予 | 余 | 与 | 容 | 揺 |
| 4D4D | 4D4F | 4D51 | 4D57 | 4D5E |
| 様 | 溶 | 用 | 要 | 抑 |
| 4D6D | 4D6E | 4D70 | 4D77 | 4D78 |
| 絡 | 落 | 乱 | 覧 | 利 |
| 4D7D | 4E22 | 4E25 | 4E28 | 4E29 |
| 理 | 裏 | 離 | 率 | 立 |
| 4E2C | 4E3B | 4E3E | 4E41 | 4E49 |
| 略 | 了 | 両 | 料 | 良 |
| 4E4C | 4E4E | 4E4F | 4E50 | 4E58 |
| 量 | 領 | 力 | 緑 | 輪 |
| 4E60 | 4E61 | 4E63 | 4E64 | 4E69 |
| 類 | 令 | 例 | 冷 | 礼 |
| 4E73 | 4E74 | 4E7D | 4F22 | 4F29 |
| 列 | 劣 | 練 | 連 | 路 |
| 4F3F | 4F40 | 4F42 | 4F43 | 4F44 |
| 録 | 論 | 和 | 話 | 歪 |
| 4F48 | | | | |
| 枠 | | | | |

# J PARAMETERS

## J.1 COMPILE PARAMERTES

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 8000 | ROMM | 16BT | | BAUD | | M30 | M02 | M99 |

(Exclusive for FANUC SYSTEM P-MODEL G)

M99 0 : M99 does not terminate compilation of a single program.
1 : M99 terminates compilation of a single program.

M02 0 : M02 does not terminate compilation of a single program.
1 : M02 terminates compilation of a single program.

M30 0 : M30 does not terminate compilation of a single program.
1 : M30 terminates compilation of a single program.

BAUD 0 : Data is transferred to the FA-WRITER at a baud rate of 4800 bps.
1 : Data is transferred to the FA-WRITER at a baud rate of 9600 bps.

16BT 0 : Data is written into the ROM module in units of 32 bits. (For Series 16)
1 : Data is written into the ROM module in units of 16 bits. (For Series 18)

ROMM 0 : The ROM cassette is used. (For Series 0)
1 : The ROM module is used. (For Series 16 or 18)

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 8010 | | | | | | | | CAP |

(Exclusive for FANUC SYSTEM P-MODEL G)

CAP 0 : The macro executor has one file.
1 : The macro executor has two files. (Conversational type)

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 9000 | LD6 | | | R1M | R512 | R256 | R128 | VRFY |

(Exclusive for FANUC SYSTEM P-MODEL G)

VRFY 0 : The ROM is not checked when the data is output to it.
1 : The ROM is checked when the data is output to it.

R128 0 : Data is output to the 64K-byte ROM cassette. (Only for FS0)
1 : Data is output to the 128K-byte ROM cassette or module. (Only for FS18/20-MA/21-TB)

R256 0 : R128 is referenced.
1 : Data is output to the 256K-byte ROM cassette or module.

R512 0 : R256 is referenced.
1 : Data is output to the 512K-byte ROM cassette or module.

R1M 0 : R512 is referenced.
1 : Data is output to the 1M-byte ROM cassette or module.

LD6 1 : Always specify 1.

**NOTE**
If both R512 and R256 are set to 1, the data is output to the 768K-byte ROM module.

| | | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|---|
| Compile parameter | 9000 | | M3MB | M2MB | M1MB | M512 | M256 | M128 | |

(Only for personal computer)

| | | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|---|
| Compile parameter | 9001 | | | | | | M4MB | | SEQN |

(Only for personal computer)

| | M4MB | M3MB | M2MB | M1MB | M512 | M256 | M128 |
|---|---|---|---|---|---|---|---|
| 4.0 MB | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3.0 MB | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 2.0 MB | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1.0 MB | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 512 KB | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 256 KB | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 128 KB | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

**NOTE**

1 For the 21-TB, always specify 128 KB.
2 For the Series 16, a 128-KB ROM-format file cannot be specified.
3 For the 21-MB, no more than 1.0 MB can be specified.

SEQN 0 : When data is output to the ROM, sequence numbers are not output to the P-CODE program.

1 : When data is output to the ROM, sequence numbers are output to the P-CODE program.

**NOTE**

When SEQN is set to 0, the P-CODE program requires a small amount of space and can be executed at high speed. However, this cannot be specified in the following cases: When GOTO is specified by a variable in the P-CODE program and when a program contains M99 and the number of the sequence (P) to which it is to be returned. (If an attempt is made to set SEQN to 0 in either of these cases, a compilation error occurs.)

Examples) GOTO #101;
M99 P100;

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 9002 | EXT1 | PWSR | DAUX | XDIL | EVF | ACL2 | ACL1 | TCAL |

TCAL 0 : Makes invalid the sub-program call with T code
1 : Makes effective the sub-program call with T code

ACL1 0 : Makes invalid call by the specific code
1 : Makes effective call by the specific code (O9004/#146)

ACL2 0 : Makes invalid call by the specific code
1 : Makes effective call by the specific code (O9005/#147)

EVF 0 : Extension P-CODE variable #20000 is the floating decimal point format
1 : Extension P-CODE variable #20000 is the fixed decimal format

XDIL 0 : Axis interlock function invalid
1 : Axis interlock function valid

DAUX 0 : Does not make the CUSTOM screen appear at power on
1 : Makes CUSTOM screen appear at power on

PWSR 0 : P CODE work number search function invalid
1 : P CODE work number search valid

EXT1 0: Extension functions invalid
1: Extension functions valid (RS-232-C control, NC program access)

**NOTE**
When EXT1 = 1, part program memory reduces by 1.63 m (In case of 21-TB or when part program memory capacity is 80 m or less).

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 9003 | | PTCR | KY20 | | | HRGR | | ONMSK |

ONMSK 0 : The O and N numbers are displayed on the CUSTOM screen.
1 : The O and N numbers are not displayed on the CUSTOM screen.

HRGR 0 : Standard mode graphic display
1 : High resolution graphic mode display (set to “1” usually)

KY20 0 : Not +20 to #8501 with decimal point input by key input variables
1 : +20 to #8501 with decimal input by key input variables

PTCR 0 : Does not output ”CR” code twice in P-CODE variable outputting
1 : Output ”CR” code twice in P-CODE variable outputting

SP_G_B, SP_G_C
00: Standard G code system
01:G code system B
10:G code system C
11:G code system C

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 9004 | CUTLG | NOP_B | | HRGC | | IMG | SP_G_C | SP_G_B |

IMG 0 : FS16-T type (Special G code)
1 : FS16-G type (Special G code)

NOP_B 0 : When no other address is specified in a block that calls a subprogram by a T or M code, the block is executed.
1 : The block is not executed.

HRGC 0 : 9” high resolusion monochrome CRT, standard
1 : 9” high resolution monochrome CRT, brightness modulation mode setting (paint)

CUTLG 0 : Does not count cutting distance
1 : Count cutting distance

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 9005 | TMACC | | | AXCLS | AX4CL | AX3CL | AX2CL | AX1CL |

AX1CL 0 : Subprogram call by the 1st address is invalid
1 : Subprogram call by the 1st address is valid

AX2CL 0 : Subprogram call by the 2nd address is invalid
1 : Subprogram call by the 2nd address is valid

AX3CL 0 : Subprogram call by the 3rd address is invalid
1 : Subprogram call by the 3rd address is valid

AX4CL 0 : Subprogram call by the 4th address is invalid
1 : Subprogram call by the 4th address is valid

AXCLS 0 : Always program O9009 is called irrespective of specified axes.
1 : The program number to be called depends on a specified axis:
Program O9031 is called when 1st axis is specified.
Program O9032 is called when 2nd axis is specified.
: :
Program O9038 is called when 8th axis is specified.

TMACC 0 : Macro call by T code is invalid
1 : Macro call by T code is valid

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 9006 | | | US19W | CNCHG | DAUXR | STDM | KEYC | DIOC |

DIOC 0 : Standard UI:G54/G55 and UO:F54/F55 are used for UI/UO(#1000-#1132) of execution macro/ conversational macro.
1 : UI:G82/G83 and UO:F84/F85 are used for UI/UO (#1000-#1132) of execution macro/ conversational macro.

KEYC 0 : When KEY switch =0, completion code #8529=254 in an access to NC program
1 : KEY switch is not cheched in an access to NC program

STDM 0 : The mode and status display is changed on the USER-1, USER-2, and USER-3 screens on which the conversational macro screen is displayed.
1 : The mode and status display is left unchanged on the USER-1, USER-2, and USER-3 screens on which the conversational macro screen is displayed.

DAUXR : The conversational macro screen is displayed when power is turned on while the system is in the emergency stop state or external reset state.
(Parameter DAUX (No. 9002, #5) must also be set to 1.)

CNCHG : Execution of the conversational macro is continued when the CUSTM key is pressed while the conversational macro screen is displayed.

US19W : The screen for the 9” CRT is displayed on the 14” CRT. (USER-1 screen)

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 9007 | | | | | | TIVR2 | TIVR1 | TTDSP |

TTDSP : The common conversational macro screen for the F16-TTA is specified.
(This setting is validated only for the HEAD-2 cassette.)
(Both TTDSP and CNCHG (No. 9006, #4) cannot be set to 1 at the same time.)

TTVR1 : Common conversational macro variables (#10000 to #19999) are specified.
(1: For variables #10000 to #19999, areas #10000 to #19999 of the other head are used.)

TTVR2 : Common conversational macro variables (#20000 to #29999) are specified.
(1: For variables #20000 to #29999, areas #20000 to #29999 of the other head are used.)

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 9008 | HRGCC | | MCARG | | AX8CL | AX7CL | AX6CL | AX5CL |

AX5CL 0 : Subprogram call by the 5th axis address is made invalid.
1 : Subprogram call by the 5th axis address is made valid.

AX6CL 0 : Subprogram call by the 6th axis address is made invalid.
1 : Subprogram call by the 6th axis address is made valid.

AX7CL 0 : Subprogram call by the 7th axis address is made invalid.
1 : Subprogram call by the 7th axis address is made valid.

AX8CL 0 : Subprogram call by the 8th axis address is made invalid.
1 : Subprogram call by the 8th axis address is made valid.

MCARG 0 : In macro calling of G/M codes, P/L/N/G is not argument.
1 : In macro calling of G/M codes, P/L/N/G is argument.

HRGCC 0 : Does not set the character display screen to brightness modulation mode.
1 : Sets the character display screen to brightness modulation mode.

| | |
|---|---|
| 9010 | M code calls sub-program O9001 |
| 9011 | M code calls sub-program O9002 |
| 9012 | M code that calls sub-program O9003 |
| 9013 | G code that calls custom macro O9010 |
| • • • | • • • |
| 9022 | G code that calls custom macro O9019 |
| 9023 | M code that calls sub-program O9020 |
| • • • | • • • |
| 9032 | M code that calls sub-program O9029 |
| 9033 | M code that calls user program |
| 9034 | G code that to cancel the modal call |
| 9035 | M code calls sub-program O9001 |
| 9036 | M code calls sub-program O9002 |

Parameters 9035 and 9036 specify the bit signal of the internal PMC relay (R area) that determines the control mode of the interlock function for a single axis direction.

Signal number (0 to 999): Specifies the number of the internal PMC relay (R area).

Signal position (0 to 7): Specifies the bit position of the signal.

Example) When parameters 9035 and 9036 are set to 900 and 7 respectively

The interlock function for a single axial direction is validated when the internal PMC relay (R900, #7) is set to 1 in the JOG or HNDL mode.

| | |
|---|---|
| 9037 | Used number of (1/100) of conversational macro exclusive variable |
| 9038 | Conversational macro execlusion program at the time of power supply input |

(CUSTOM screen)

| | |
|---|---|
| 9039 | Auxility macro execution program number |
| 9040 | Conversational macro execution program number at the time of power on |

(MENU screen)

| | |
|---|---|
| 9041 | Conversational macro execution program number at the time of power on |

(MACRO screen)

| | |
|---|---|
| 9042 | M code call by area specification, lower limit M code |

| | |
|---|---|
| 9043 | M code call by area specification, upper limit M code |
| 9044 | Used number of extension conversational macro variables |
| 9045 | Starting G code in G code calls of ,range specification |
| 9046 | Number of G codes in G code calls of ,range specification |
| 9047 | Starting O number in G code calls of ,range specification |
| 9048 | Distance by which the graphics coordinate system is shifted on the conversational macro screen (in the X direction) |
| 9049 | Distance by which the graphics coordinate system is shifted on the conversational macro screen (in the Y direction) |

These parameters specify the distance by which the graphics coordinate system is shifted on the conversational macro screen in units of dots.

## J.2 EXECUTOR PARAMETER

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 9000 | L2R | | MKG | RSC | EXS | STP | NDP | SQN |

SQN 0 : Displays program number and sequence number of called user program during execution of a recorded program.

1 : Displays program number of recorded program and sequence number during execution of a recorded program.

NDP 0 : Does not display variables for P-CODE program.

1 : Displays variables for P-CODE program.

STP 0 : Execute the conversational macro program.

1 : Stops execution of conversational macro program. (”1” is set by break function automatically)

EXS 0 : Stop if feed hold acts, during macro statement execution at execution level.

1 : Execute macro statement till the next NC statement has come with feed hold, during macro statement execution at execution level.

RSC 0 : Common variables #100-#149 do not clear to <vacant> if NC is reset.

1 : Common variables #100-#149 are cleared to <vacant> if NC is reset.

MKG : Be sure to set to 0.

L2R 0 : Display conversational / auxiliary macro and screen during execution macro operation.

1 : Does not display conversational / auxiliary macro and screen during execution macro operation.

| | |
|---|---|
| 9002 | Conversational macro break program number |

| | |
|---|---|
| 9003 | Conversational macro break sequence number |

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 9010 | MA8 | MA7 | MA6 | MA5 | MA4 | MA3 | MA2 | MA1 |

MA1 0 : Does not mask a first axis address macro call.

1 : Masks a first axis address macro call.

MA2 0 : Does not mask a second axis address macro call.

1 : Masks a second axis address macro call.

MA3 0 : Does not mask a third axis address macro call.

1 : Masks a third axis address macro call.

MA4 0 : Does not mask a fourth axis address macro call.

1 : Masks a fourth axis address macro call.

MA5 0 : Does not mask a fifth axis address macro call.

1 : Masks a fifth axis address macro call.

MA6 0 : Does not mask a sixth axis address macro call.

1 : Masks a sixth axis address macro call.

MA7 0 : Does not mask a seventh axis address macro call.

1 : Masks a seventh axis address macro call.

MA8 0 : Does not mask an eighth axis address macro call.

1 : Masks an eighth axis address macro call.

> **NOTE**
> When compilation parameter bit 4 of No. 9005 (AXCLS) is set to 1, MA1 masks O9031, MA2 masks O9032, MA3 masks O9033, and so on.

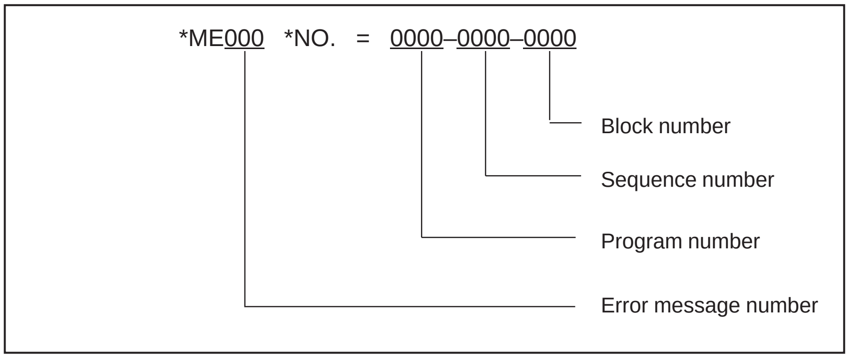
| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 9011 | | | | | | | | MTC |

MTC 0 : Does not mask a T code execution macro call.
1 : Masks a T code execution macro call.

# K ERROR CODES

# K.1 COMPILER ERROR CODES (SYSTEM P)

(1) Display of error codes
The error codes are displayed as follows when an error occurs.




If the program does not contain any sequence number, the program number is not shown in the error message.

The program number, sequence number and block number are not displayed in error messages with numbers greater than 100.

(2) Table of error codes and their meaning

| No. | Explanation |
|---|---|
| 001 | The program number exceeds the maximum registered number. Up to 400 pieces loaded. |
| 002 | No program available. |
| 011 | The block delete address includes a decimal point. |
| 012 | The block delete address is out of the 1 to 9 range |
| 013 | The program has a program number other than the head of program. |
| 014 | The block has a sequence number other than the head of program. |
| 015 | The NC statement contains an error. |
| 016 | The macro statement ends with an other code than EOB. |
| 017 | The equal sign of the macro statement cannot be found. |
| 018 | The nesting of DO exceeds 3 levels |
| 019 | The relational operator in the conditional expression is not found. |
| 020 | No GOTO after IF. |
| 021 | The ']' of the IF [<conditional expression>] is not found. |
| 022 | There is a code other than EOB after GOTOn. |
| 023 | There is a code than EOB after DOm. |
| 024 | There is a code than EOB after ENDm. |
| 025 | The END identification number does not correspond to that of DO. |
| 030 | No END found for DO. |
| 031 | No DO found after WHILE. |
| 032 | The ']' of the WHILE [<conditional expression>] is not found. |
| 033 | It is unclear whether the block is a NC statement or a macro statement. |
| 034 | No DO found for END. |
| 035 | The program number in the directory and actual number used in the program do not correspond. |
| 036 | No program number in the program heading. |
| 041 | Nesting of brackets exceeds 5 levels. |
| 042 | The ']' of the # [<expression>] is not found. |

| No. | Explanation |
|---|---|
| 043 | The ’]’ of the [<expression>] is not found. |
| 044 | The second ’]’ of the ATAN [<expression>]/[<expression>] is not found. |
| 045 | The ’]’ of the ATAN [<expression>]/[<expression>] is not found. |
| 046 | The first ’]’ of the ATAN [<expression>]/[<expression>] is not found. |
| 047 | The ’]’ of the function [<expression>] is not found. |
| 048 | The <expression> format contains an error. |
| 049 | The left part of <expression> of the substitution statement contains an error. |
| 050 | The <expression> of the <address>[<expression>], <address> - [<expression>] or GOTO [<expression>] format contains an error. |
| 051 | The numeral contains more than 8 figures. |
| 052 | Other codes than numeric codes follow the decimal point. |
| 053 | The variable address of the macro variable consists of more than 6 digits. |
| 054 | No figure or ’[’ following #. |
| 055 | The program number consists of more than 4 figures. |
| 056 | The sequence number contains more than 4 figures. |
| 057 | The ’[’ of function [<expression>] is not found. |
| 058 | The ’[’ of IF [<expression>] or WHILE [<expression>] is not found. |
| 059 | m of DOm or ENDm consists of more than 1 digit. |
| 060 | DOm or ENDm is out of the 1 to 3 range. |
| 061 | Other codes than numeric codes follow DO or END. |
| 062 | Other alphabetical string than control directive or function found. |
| 063 | Alphabetical string of more than 5 characters found. |
| 064 | No EOR at the end of the program. |
| 065 | Inappropriate code found in the program. |
| 070 | Character string exceeds 255 characters. |
| 071 | Inner code consists of more than 4 figures. |
| 072 | Inner code is not in hexadecimal format. |
| 073 | Non-displayable system code is specified. |
| 074 | Not end with ’*)’ for character string starting with ’(*’. |
| 075 | ’(’ and ’)’ hold characters that are not allowed. |
| 081 | The transient variable area used by the macro executor is not available. Too many addresses in the <expression> of 1 block of a NC statement. |
| 082 | The branch point for the GOTO statement is too large. Add a parameter to set the branch point as 4 bytes or reduce the program size. |
| 083 | The number of addresses contained in 1 block of the NC statement exceeds 50. |
| 084 | The variable No. of the macro variable has more than 6 digits. |
| 085 | The variable No. of the macro variable is negative. |
| 086 | The variable No. of the macro variable contains a decimal point. |
| 087 | More than 4 digits following the GOTO statement. |
| 088 | The figures following the GOTO statement contain a decimal point. |
| 091 | The number of GOTO statements in one block directly pointing to a sequence number, exceeds the limit (200). |
| 092 | The sequence number indicated as the branch point of the GOTO statement is not found. |
| 093 | The number of WHILE statements in one program exceeds the limit (200). |
| 094 | There are several sequence numbers for the branch point of the GOTO statement. |
| 100 | The macro executor is not read into memory. |

| No. | Explanation |
|---|---|
| 101 | The ROM module has caused a memory overflow. |
| 102 | 256  KByte ROM module cannot be used  with  this  macro executor. |
| 111 | A time error occurred while waiting for answer from  the FA writer. |
| 112 | A parity error occurred while waiting for answer from the FA writer. |
| 113 | An overflow error occurred while waiting for answer  from the FA writer. |
| 114 | Framing error occurred while waiting for answer from  the FA writer. |
| 115 | FA writer is not READY or cable is not connected. |
| 116 | Error occurred during transmittion to the FA writer. |
| 117 | Parity error occurred while sending to the FA writer. |
| 118 | ROM is not erased. |
| 119 | ROM write error occurred. |
| 120 | ROM verify error occurred. |
| 121 | ROM module is not installed. |
| 122 | Other than order made macro ROM module is installed. |
| 123 | Wrong FA writer version number. |
| 124 | Error occurred in FA writer. |
| 125 | The address or length instruction for the FA writer  is uneven. |
| 126 | The written data exceeds the capacity of  the  installed ROM module. |
| 127 | The FA writer is not set up properly. |
| 128 | Address setting error of the FA writer. |
| 129 | Inverse installment of ROM. |
| 131 | Memory write error. |
| 132 | Memory read error. |
| 140 | File open parameter error. |
| 141 | Floppy disk hard error. |
| 142 | File not found. |
| 143 | Wrong file format. |
| 144 | Floppy disk already in use. |
| 145 | File is protected. |
| 146 | File name already in use. |
| 147 | Password error. |
| 148 | File size overflow. |
| 149 | File number overflow. |
| 150 | File closed error. |

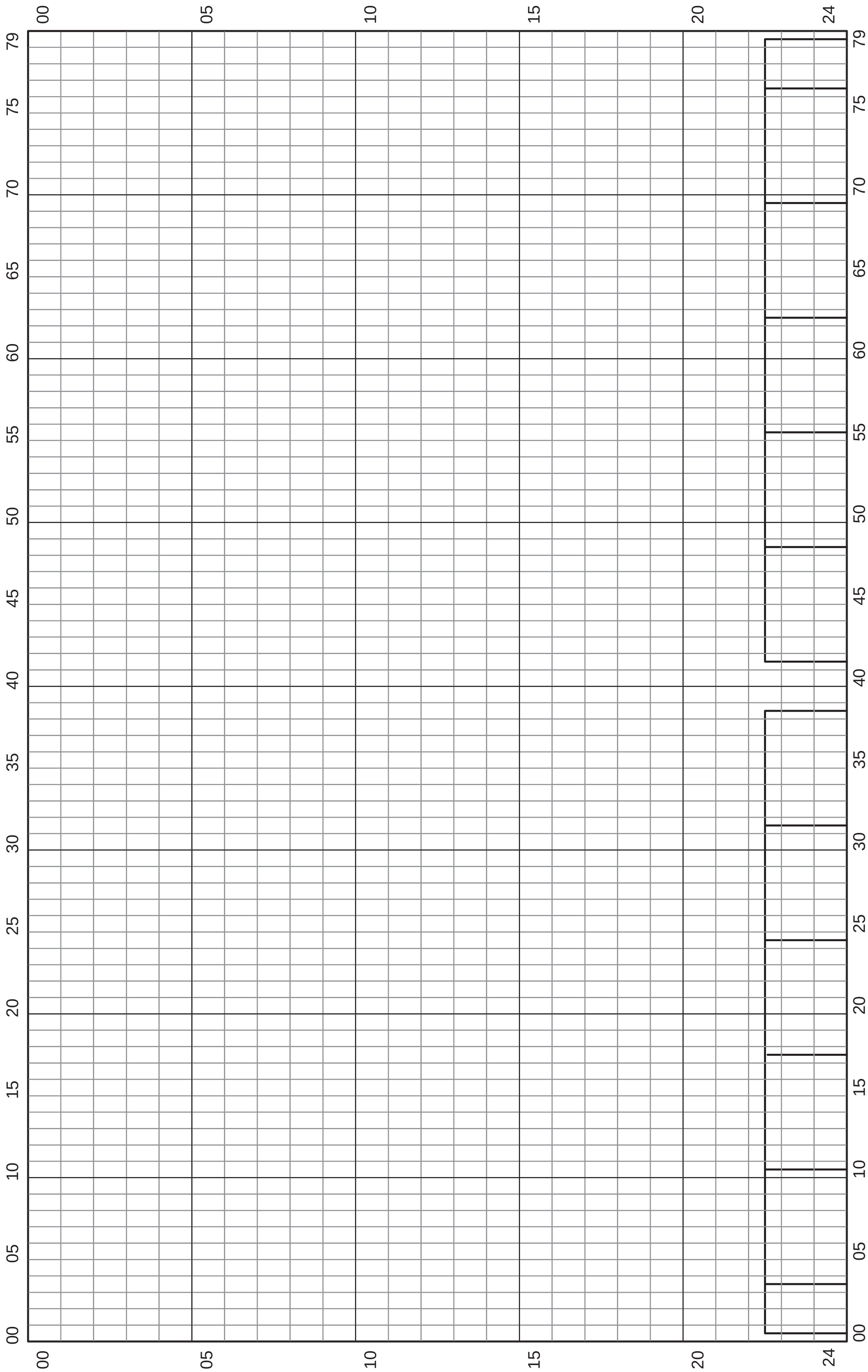
## K.2 EXECUTOR ERROR CODE

The following is an explanation of the supplementary P/S error codes that may occur at the time of execution of P-CODE program.

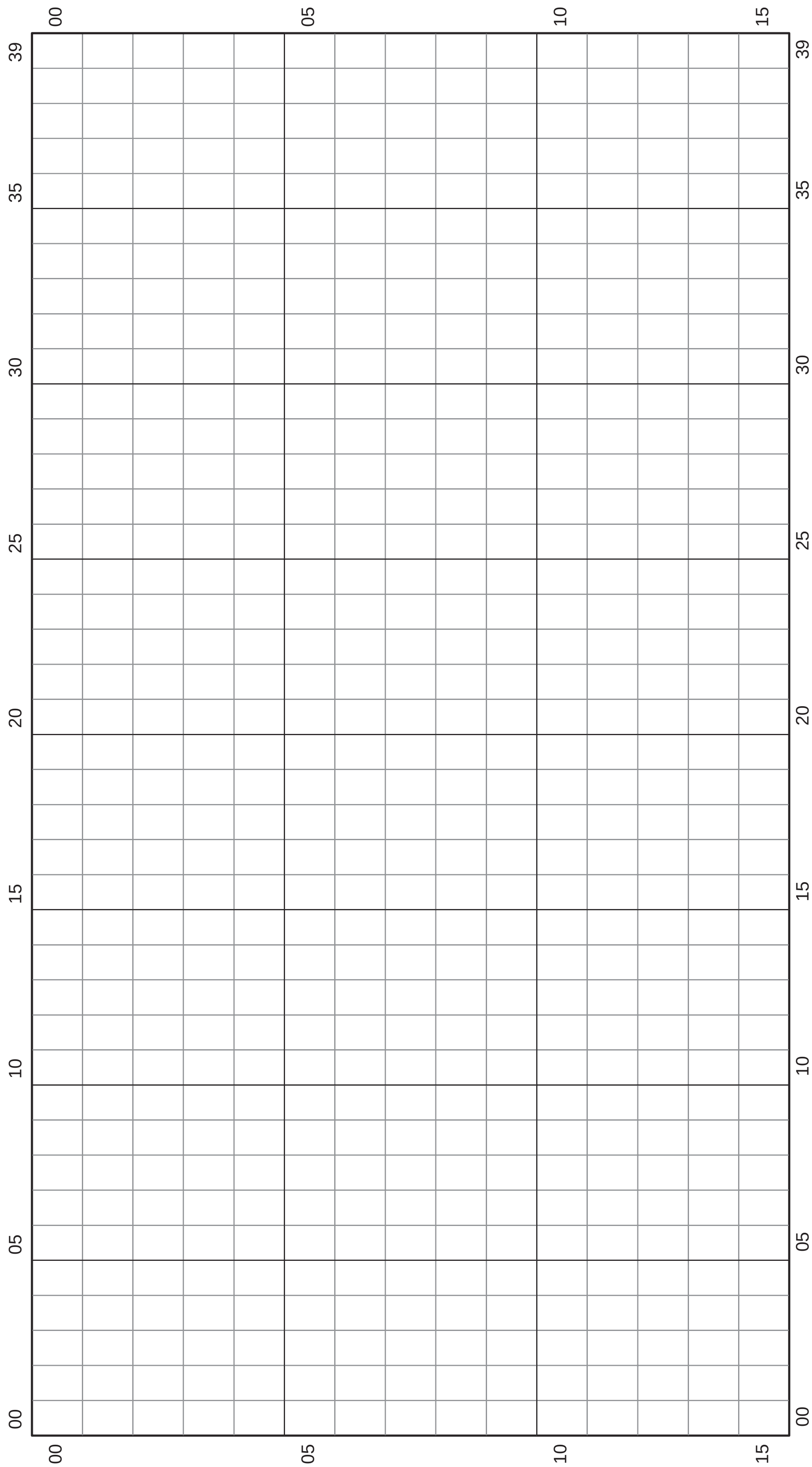
| Code | Explanation |
|---|---|
| 110 | The absolute value of the data of the fixed decimal point display exceeds the allowable range. |
| 111 | The index of the data of the floating decimal point exceeds the allowable range. |
| 112 | The divisor is 0. |
| 115 | Pointing to the value of an undefined variable address. |
| 116 | The left part of the substitution statement consists of an illegal variable. |
| 119 | The SQRT parameter is a negative value, the BCD parameter is a negative value or the BIN parameter contains values others than 0 to 9. |

# L CHARACTER COORDINATE SYSTEM

(1) For 12 soft keys display



(2) For 7soft keys display



# M SCREEN DISPLAY ON A VGA GRAPHICS DISPLAY UNIT (SUPPORTED BY Super CAP EXECUTOR ONLY)

# M.1 DISPLAY COLOR SPECIFICATION EXTENSION

## M.1.1 Overview

When using a VGA graphics display unit, the user can choose any of sixteen colors for character display and graphics display.

Moreover, the user can choose from sixteen colors for the character display background.

The initial color settings on the CUSTOM screen are as follows: color palette 7 for character display/graphics display, color palette 0 for the background, and non–blinking.

G240 P_C_L_;

P: Character display/graphics display color specification
When one of the following values is specified with a minus sign (–) prefixed, characters are displayed in reverse video.

| | | | |
|---|---|---|---|
| =0 | . . Color of color palette 0 | Default : | Black |
| =1 | . . Color of color palette 1 | | Red |
| =2 | . . Color of color palette 2 | | Green |
| =3 | . . Color of color palette 3 | | Yellow |
| =4 | . . Color of color palette 4 | | Blue |
| =5 | . . Color of color palette 5 | | Purple |
| =6 | . . Color of color palette 6 | | Peacock blue |
| =7 | . . Color of color palette 7 | | White |
| =8 | . . Color of color palette 8 | | Light black |
| =9 | . . Color of color palette 9 | | Light red |
| =10 | . . Color of color palette 10 | | Light green |
| =11 | . . Color of color palette 11 | | Light yellow |
| =12 | . . Color of color palette 12 | | Light blue |
| =13 | . . Color of color palette 13 | | Light purple |
| =14 | . . Color of color palette 14 | | Light peacock blue |
| =15 | . . Color of color palette 15 | | Light white |

C: Specification of the background color

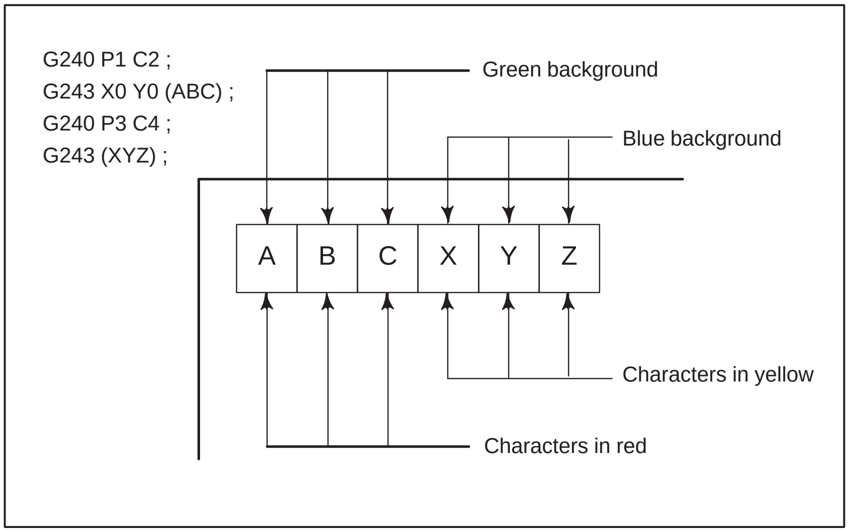
| | | | |
|---|---|---|---|
| =0 | . . Color of color palette 0 | Default : | (Same as above) |
| =1 | . . Color of color palette 1 | | |
| =2 | . . Color of color palette 2 | | |
| =3 | . . Color of color palette 3 | | |
| =4 | . . Color of color palette 4 | | |
| =5 | . . Color of color palette 5 | | |
| =6 | . . Color of color palette 6 | | |
| =7 | . . Color of color palette 7 | | |
| =8 | . . Color of color palette 8 | | |

=9 . . Color of color palette 9
=10 . . Color of color palette 10
=11 . . Color of color palette 11
=12 . . Color of color palette 12
=13 . . Color of color palette 13
=14 . . Color of color palette 14
=15 . . Color of color palette 15

L: Specification of blinking

=0 . . Non–blinking
=1 . . Blinking

The following specification displays the colors shown below:




When only addresses P and C are specified, the specification of 0 at address L is assumed.

G240 P1 ; Character color: Color of color palette 1
Background color: (No change)
Blinking: Non–blinking

G240 C1 ; Character color: (No change)
Background color: Color of color palette 1
Blinking: Non–blinking

**NOTE**
Color palettes cannot be set using the executor. Those set using the CNC are used.

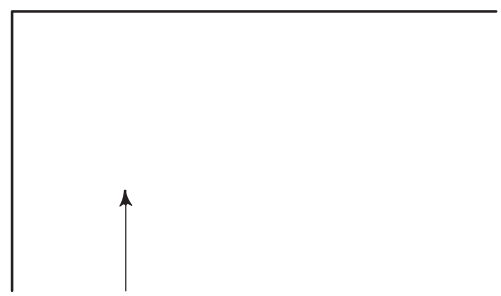
# M.2 SCREEN ERASURE EXTENSION

## M.2.1 Overview

When a VGA graphics display unit is used, the screen is erased using color palette 7 for character display and a selected color for the background.

If no of background color has been specified on the CUSTOM screen, color palette 0 is used.

```
G240 P1 C2 ;
G202 P2 ;
```



The screen is erased using color palette 7 for characters and color palette 2 for the background.

**NOTE**

For display units other than a VGA graphics display, white is used for screen erasure. This corresponds to color palette 7, so that color palette 7 is used for erasure on a VGA graphics display unit.

# M.3 EXTENSION OF NUMBER OF DISPLAY LINES

## M.3.1 Overview

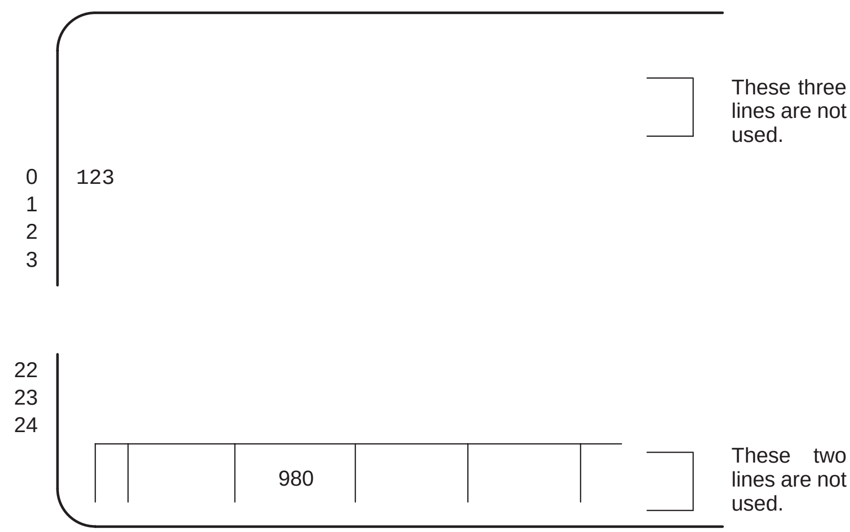
When a VGA graphics display unit having twelve soft keys is used, and bit 2 (CM30) of compile parameter No. 9009 is set to 1, up to 30 lines including the upper three lines and lower two lines, which are not usually used, can be used to display characters.

When a VGA graphics display unit having seven soft keys is used, up to 19 lines including the upper two lines and lower one line which are not usually used, can be used to display characters.

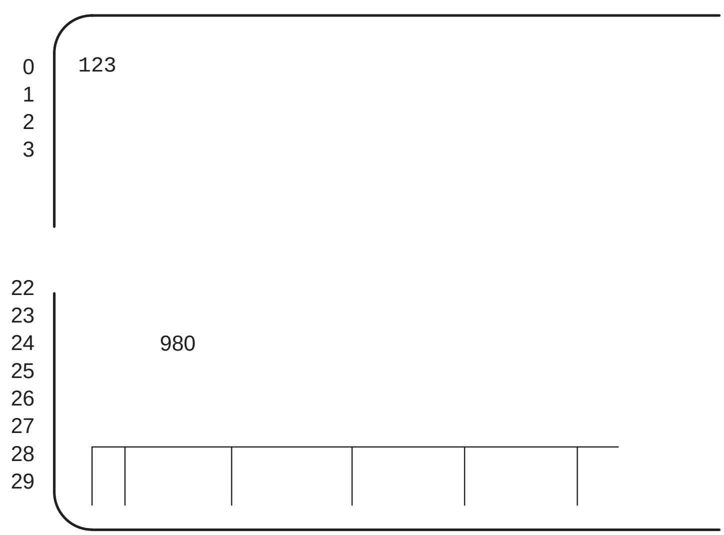
When the following command is issued with a VGA graphics display unit, the usable lines can be increased as shown below:

G243 X0 Y0 (123) ;
G243 X4 Y24 (980) ;

When bit 2 of compile parameter No. 9009 = 0




When bit 2 of compile parameter No. 9009 = 1



## M.3.2 Compile Parameters

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 9009 | | | | | | CM30 | LM30 | |

LM30 0 : The full 30–line screen is not used for VGA display. (Super CAP)

1 : The full 30–line screen is used for VGA display. (Super CAP)

CM30 0 : The full 30–line screen is not used for VGA display. (CUSTOM)

1 : The full 30–line screen is used for VGA display. (CUSTOM)

**NOTE**

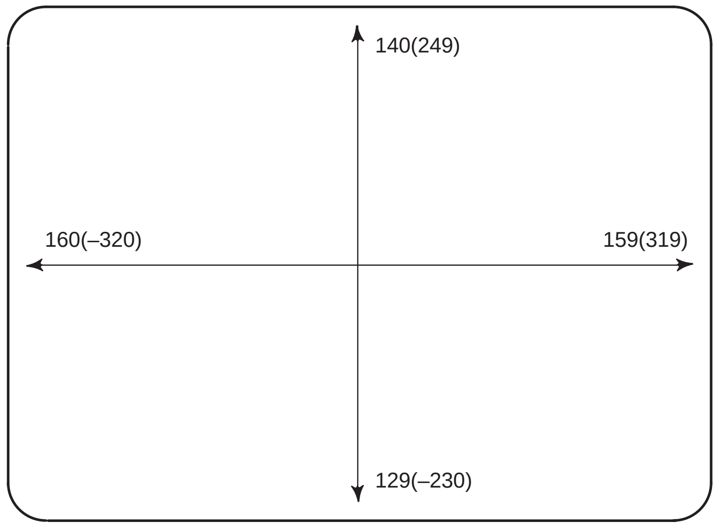
LM30 is enabled when a command related to executor screen display is used on a screen other than the CUSTOM screen.

# M.4 GRAPHICS COORDINATE SYSTEM

## M.4.1 Overview

When a VGA graphics display unit having twelve soft keys is used, the X coordinate of the graphics coordinate system ranges from –320 to 319 (from left to right), while the Y coordinate ranges from –232 to 247 (from bottom to top). When a VGA graphics display unit having seven soft keys is used, the X coordinate ranges from –160 (–320) to 159 (319) (from left to right), while the Y coordinate ranges from –129 (–232) to 140 (247) (from bottom to top).

Display unit having seven soft keys




High–resolution mode in parentheses

Display unit having twelve soft keys


> **NOTE**
> For a display unit having twelve soft keys, the standard (low–resolution) display mode is not supported.

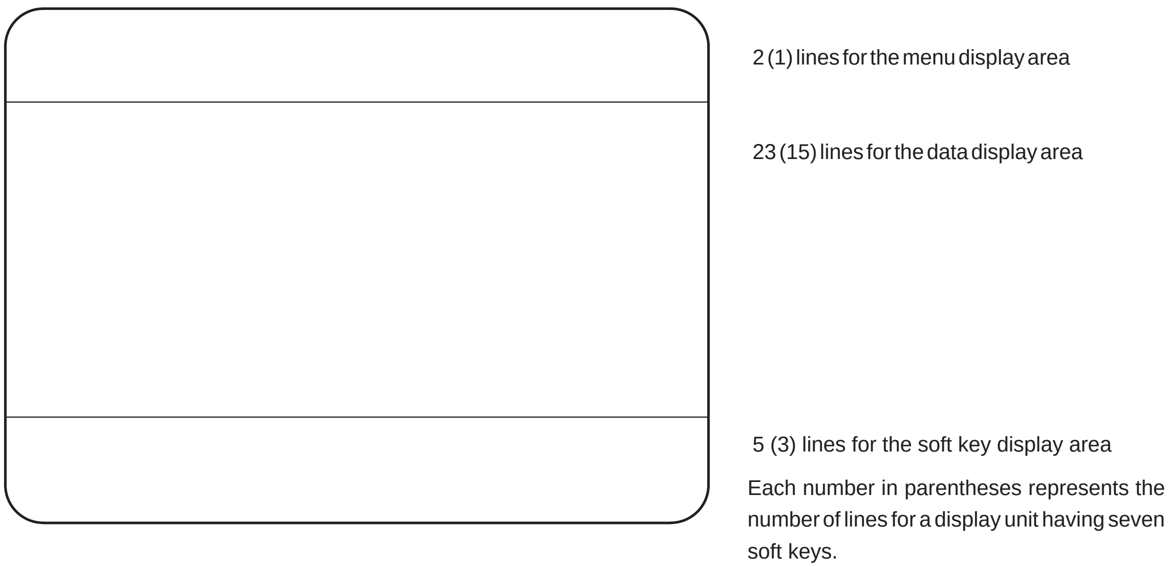
# N DISPLAY WITH A BACKGROUND COLOR ON THE CUSTOM SCREEN

## N.1 OVERVIEW

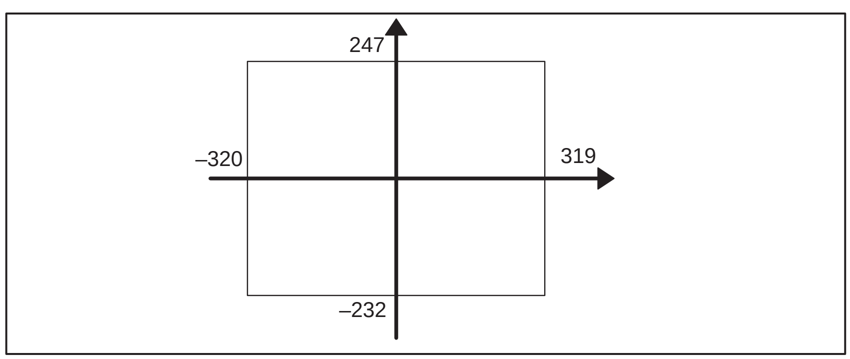
When a VGA graphics display unit is used with the Series 16i/18i/21i, display with a background color can be provided on the conversational macro (CUSTOM) screen by setting bit 0 (VGAR) of compile parameter No. 9100 to 1.
Display with a background color can be provided in the following areas:

Display with twelve soft keys: 30 lines (vertically), 80 characters (horizontally)

Display with seven soft keys: 19 lines (vertically), 40 characters (horizontally)




Display with a background color is provided using the graphics display. When graphics is specified in the same position, display with a background color is overwritten. Similarly, display with a background color can overwrite the specified graphics display.
When a screen erase command (G202) is specified, graphics display is erased using color palette 15.
When a dot coordinate system is used, the number of dots is 640 x 480, regardless of whether a display unit having twelve or seven soft keys is used. The coordinate system is as follows:




## N.2 COMMAND FOR DISPLAY WITH A BACKGROUND COLOR

G250 P_<parameter>

P_: Specifies an item number.

<parameter>: Specifies a parameter for each item.

## N.3 ITEMS OF THE COMMAND FOR DISPLAY WITH A BACKGROUND COLOR (P_)

| Item (P_) | Description | Parameter |
|---|---|---|
| 000 | Clears the screen with a background color. | None |
| 001 | Clears the data display area only. | None |
| 002 | Clears the screen background with a background color | None |
| 003 | Clears only the background of the data display area. | None |
| 010 | Displays a convex group frame. | X_ Y_ I_ J_ |
| 011 | Displays a concave group frame. | X_ Y_ I_ J_ |
| 015 | Displays a key–in line frame. | X_ Y_ |
| 018 | Displays a frame for 9 selected window(*). | X_ Y_ |
| 019 | Displays a frame for an unselected window(*). | X_ Y_ |
| 020 | A window for window frame mode(*). | X_ Y_ R_ |
| 021 | Selects a window for window frame mode(*). | R_ |
| 022 | Displays a frame for a selected window in window frame mode(*). | R_ |
| 023 | Displays a frame for an unselected window in window frame mode(*). | R_ |
| 024 | Displays a frame background for a selected window in window frame mode(*). | R_ |
| 025 | Displays a frame background for an unselected window in window frame mode(*). | R_ |
| 030 | Displays soft key nonselection state. | None |
| 031 | Displays the state of soft key pressing. | R_ (B_) |
| 040 | Sets a graphics palette (1 palette). | R_ A_ B_ C_ |
| 041 | Sets a character palette (1 palette). | R_ A_ B_ C_ |
| 042 | Sets graphics/character palettes. | R_ |
| 043 | Sets graphics palettes (all palettes). | R_ |
| 044 | Sets character palettes (all palettes). | R_ |
| | | |

The items marked with an asterisk (*) can be specified only for the display unit having twelve soft keys.

A screen background with a background color corresponds to a graphics display plane. On the screen, a graphics display plane and character display plane are overlaid on each other. For display with a background color, a graphics display plane is used.

In window frame mode, display coordinates are based on the frame of a selected/unselected window. This means that, for character display, the upper–left point of each window frame serves as the coordinates (0,0). However, this does not affect the graphics display coordinates.

## N.4 DETAILS OF ITEMS OF THE COMMAND FOR DISPLAY WITH A BACKGROUND COLOR (P_)

Items (000, 001, 002, 003)

The items (000 and 002) clear the display with a background color and character display.

The items (001 and 003) clear only the display with a background color.

Items (010 and 011)

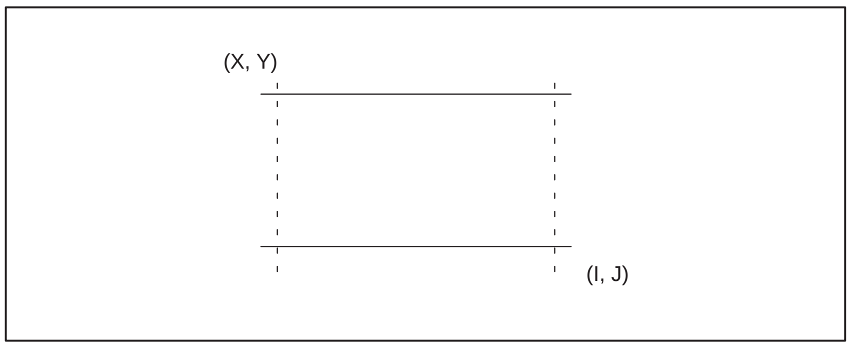
X_: Frame upper–left point (X–axis)

Y_: Frame upper–left point (Y–axis)

I_: Frame lower–right point (X–axis)

J_: Frame lower–right point (Y–axis)

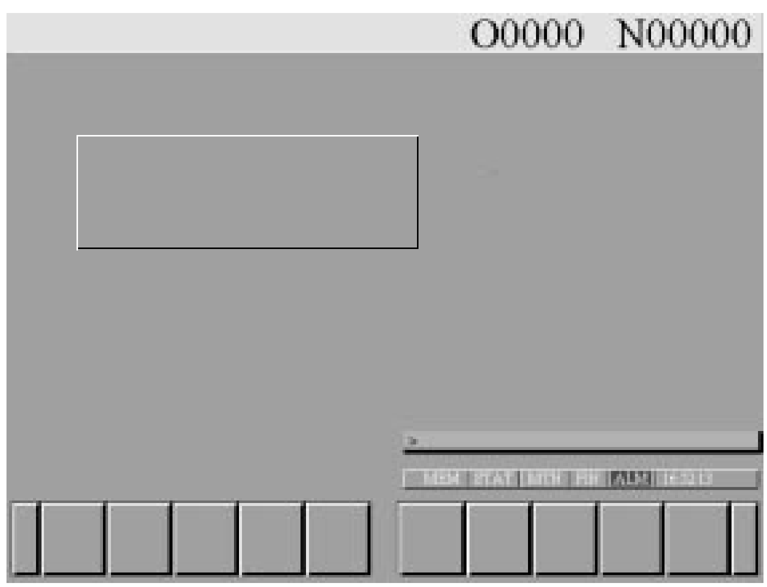
The X, Y, I, and J points represent coordinates for character display.

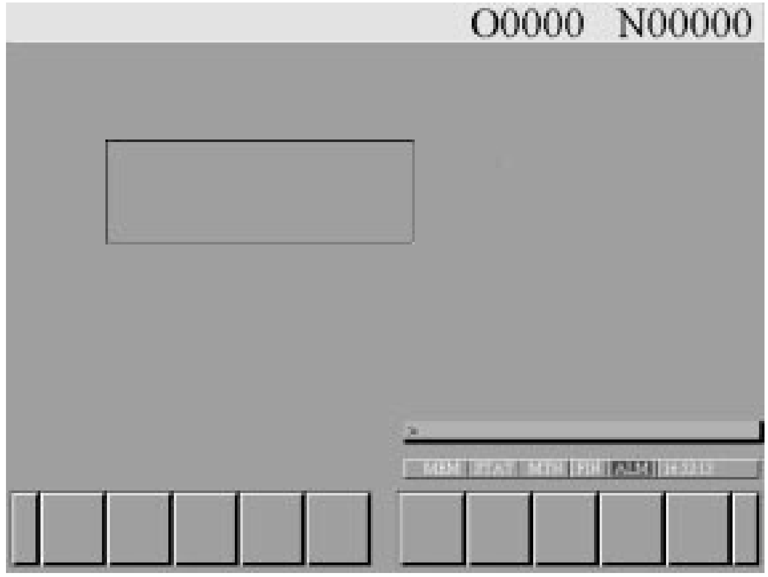



The above parameters specify a rectangular frame.

When a specified frame is too large to be displayed within the screen, the command is ignored.

Only display with a background color is supported.






Item (015)

X_: Frame start point (X–axis)

Y_: Frame start point (Y–axis)

The X and Y points represent the coordinates for character display.

A fixed frame size is used: 40 characters along the X–axis, and one line along the Y–axis. When a frame of this size cannot be displayed within the display area of the screen, the command is ignored.
Only display with a background color is supported.

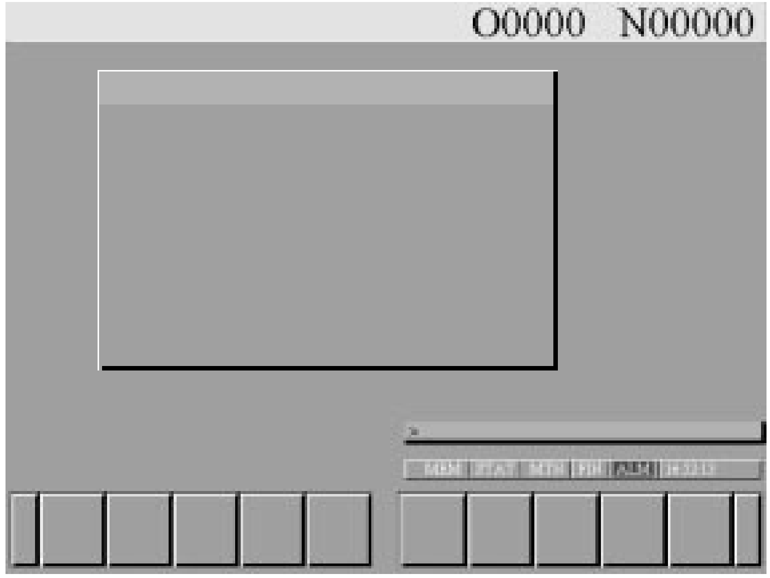
Items (018, 019) (Usable with a display unit having twelve soft keys)

X_: Frame start point (X–axis)

Y_: Frame start point (Y–axis)

The X and Y points represent the coordinates for character display.

A fixed frame size is used: 41 characters along the X–axis, and 14 lines along the Y–axis. When a frame of this size cannot be displayed within the display area of the screen, the command is ignored.




Item (020) (Usable with a display unit having twelve soft keys)

R_: Frame number (1 to 3)

X_: Frame start point (X–axis)

Y_: Frame start point (Y–axis)

The X and Y points represent the coordinates for character display.

A fixed frame size is used: 41 characters along the X–axis, and 14 lines along the Y–axis. When a frame of this size cannot be displayed within the display area of the screen, the command is ignored.

Item (021) (Usable with a display unit having twelve soft keys)

R_: Selection number of a frame registered with Item (020)

A frame registered with Item (020) is selected, but is not displayed.

Item (022, 024) (Usable with a display unit having twelve soft keys)

R_: Selection number of a frame registered with Item (020)

A frame registered with Item (020) is displayed for a selected window.

Item (020) clears the character display from within the area.

Items (023 and 025) (Usable with a display unit having twelve soft keys)

R_: Selection number of a frame registered with Item (020)

A frame registered with Item (020) is displayed for an unselected window frame.

Item (023) clears the character display from within the area.

Item (031)

R_: Soft key number

When a display unit having seven soft keys is used

1 = Selects soft key 1

2 = Selects soft key 2

3 = Selects soft key 3

4 = Selects soft key 4

5 = Selects soft key 5

When a display unit having twelve soft keys is used

1 = Selects soft key 1

2 = Selects soft key 2

3 = Selects soft key 3

4 = Selects soft key 4

5 = Selects soft key 5

6 = Selects soft key 6

7 = Selects soft key 7

8 = Selects soft key 8

9 = Selects soft key 9

10 = Selects soft key 10

When a display unit having twelve soft keys is used, one of soft keys 1 through 5, and one of soft keys 6 through 10 can be selected. When two soft keys are selected, address R is used to select one soft key, while address B is used to select the other soft key.

G250 P31 R2 B8 This command selects soft key 2 and soft key 8.

When 0 is specified in R, the command has the effect of Item (030).

Items (040, 041)

R_: Color palette number (0 to 15)

A_: R value (0 to 15)

B_: G value (0 to 15)

C_: B value (0 to 15)

Item (042)

R_: Color palette setting selection number (0 to 1)

R = 0 Standard color for a screen with a background color

| | R value | G value | B value |
|---|---|---|---|
| Graphics | | | |
| Color palette 0 | 0 | 0 | 0 |
| Color palette 1 | 15 | 0 | 0 |
| Color palette 2 | 0 | 15 | 0 |
| Color palette 3 | 15 | 15 | 0 |
| Color palette 4 | 0 | 0 | 15 |
| Color palette 5 | 15 | 0 | 15 |
| Color palette 6 | 0 | 15 | 15 |
| Color palette 7 | 15 | 15 | 15 |
| Color palette 8 | 0 | 0 | 15 |
| Color palette 9 | 0 | 11 | 11 |
| Color palette 10 | 15 | 15 | 15 |
| Color palette 11 | 10 | 9 | 9 |
| Color palette 12 | 15 | 15 | 15 |
| Color palette 13 | 12 | 11 | 11 |
| Color palette 14 | 4 | 4 | 4 |
| Color palette 15 | 12 | 11 | 11 |
| Character | | | |
| Color palette 0 | 0 | 0 | 0 |
| Color palette 1 | 8 | 0 | 0 |
| Color palette 2 | 0 | 8 | 0 |
| Color palette 3 | 8 | 8 | 0 |
| Color palette 4 | 15 | 15 | 0 |
| Color palette 5 | 15 | 0 | 15 |
| Color palette 6 | 0 | 8 | 8 |
| Color palette 7 | 3 | 1 | 1 |
| Color palette 8 | 15 | 15 | 15 |
| Color palette 9 | 13 | 13 | 13 |
| Color palette 10 | 12 | 12 | 12 |
| Color palette 11 | 11 | 11 | 11 |
| Color palette 12 | 10 | 10 | 10 |
| Color palette 13 | 9 | 9 | 9 |
| Color palette 14 | 8 | 8 | 8 |
| Color palette 15 | 7 | 7 | 7 |

R = 1 Standard color for a screen without a background color

| | R value | G value | B value |
|---|---|---|---|
| Graphics | | | |
| Color palette 0 | 0 | 0 | 0 |
| Color palette 1 | 15 | 0 | 0 |
| Color palette 2 | 0 | 15 | 0 |
| Color palette 3 | 15 | 15 | 0 |
| Color palette 4 | 0 | 0 | 15 |
| Color palette 5 | 15 | 0 | 15 |
| Color palette 6 | 0 | 15 | 15 |
| Color palette 7 | 15 | 15 | 15 |
| Color palette 8 | 0 | 0 | 15 |
| Color palette 9 | 0 | 11 | 11 |
| Color palette 10 | 15 | 15 | 0 |
| Color palette 11 | 11 | 9 | 9 |
| Color palette 12 | 15 | 15 | 15 |
| Color palette 13 | 12 | 11 | 11 |
| Color palette 14 | 4 | 4 | 4 |
| Color palette 15 | 12 | 11 | 11 |
| Character | | | |
| Color palette 0 | 0 | 0 | 0 |
| Color palette 1 | 15 | 0 | 0 |
| Color palette 2 | 0 | 15 | 0 |
| Color palette 3 | 15 | 15 | 0 |
| Color palette 4 | 0 | 0 | 15 |
| Color palette 5 | 15 | 0 | 15 |
| Color palette 6 | 0 | 15 | 15 |
| Color palette 7 | 15 | 15 | 15 |
| Color palette 8 | 14 | 14 | 14 |
| Color palette 9 | 13 | 13 | 13 |
| Color palette 10 | 12 | 12 | 12 |
| Color palette 11 | 11 | 11 | 11 |
| Color palette 12 | 10 | 10 | 10 |
| Color palette 13 | 9 | 9 | 9 |
| Color palette 14 | 8 | 8 | 8 |
| Color palette 15 | 7 | 7 | 7 |

## N.5 COMPILE PARAMETERS

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 9100 | MSFT | C9WN | DLMT | VKLN | | | VGCL | VGAR |

VGAR 0 : Does not display the CUSTOM screen with a background color.
1 : Displays the CUSTOM screen with a background color.

VGCL 0 : Does not display the CUSTOM screen with a background in a background color.
1 : Displays the CUSTOM screen with a background in a background color.

VKLN 0 : Does not display a key–in line background.
1 : Displays a key–in line background.

DLMT 0 : Does not confine display with a background color to within the data area.
1 : Confines display with a background color to within the data area.

C9WN 0 : Does not perform character display coordinate correction in seven–soft–key window display.
1 : Performs character display coordinate correction in seven–soft–key window display.

MSFT 0 : Displays a soft key frame on the CUSTOM screen.
1 : Does not display a soft key frame on the CUSTOM screen.

# N.6 RELATED ITEMS

## N.6.1 Related Compile Parameters

Note the following related compile parameters for display.

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 9002 | | | DAUX | | | | | |

DAUX = 0 : The CUSTOM screen is not displayed at power–up.
= 1 : The CUSTOM screen is displayed at power–up.

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 9003 | | | | | | HRGR | | ONMSK |

ONMSK = 0 : The O and N numbers are displayed on the CUSTOM screen.
= 1 : The O and N numbers are not displayed on the CUSTOM screen.

HRGR = 0 : Graphic display is performed in standard mode.
= 1 : Graphic display is performed in high resolution mode (only for a window displayed with seven soft keys).

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 9006 | | NNUM | US19W | | DAUXR | STDM | | |

STDM = 0 : The mode or status is displayed on the CUSTOM screen.
= 1 : The mode or status is not displayed on the CUSTOM screen.

DAUXR = 0 : The CUSTOM screen is not displayed when the system is in emergency stop status at power–up.
= 1 : The CUSTOM screen is displayed when the system is in emergency stop status at power–up. (Set the DAUX bit (bit 5 of parameter 9002) to 1 as well.)

US19W = 0 : When display with 12 soft keys is selected, the CUSTOM screen (USER1) is not displayed as a window with seven soft keys.
= 1 : When display with 12 soft keys is selected, the CUSTOM screen (USER1) is displayed as a window with seven soft keys.

NNUM = 0 : When data input control is valid for the CUSTOM screen, the NUM prompt is displayed.
= 1 : When data input control is valid for the CUSTOM screen, the NUM prompt is not displayed.

| | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |
|---|---|---|---|---|---|---|---|---|
| 9007 | | | | | US19WK | | | |

US19WK = 0 : When display with 12 soft keys is selected, the position of the key–in line is not changed in a window with seven soft keys.

= 1 : When display with 12 soft keys is selected, the position of the key–in line is changed in a window with seven soft keys.

## N.6.2 Conversational/ Auxiliary Macro Alarms

For the command for display with a background color, the following conversational/auxiliary macro alarms are added:

| Alarm No. | Description |
|---|---|
| 200 | The environment does not support the use of the function for display with a background color (G250). |
| 201 | Address P is not specified with the function for display with a background color (G250). |
| 202 | Address P, specified with the function for display with a background color (G250), is incorrect. |
| 203 | A parameter other than address P is specified incorrectly with the function for display with a background color (G250). |

## N.6.3 Graphics Display Screen Erasure

Graphics display is used for display with a background color. When character display or graphics display is erased using the screen erase command (G202), the background of the display with a background color may be erased, or the coordinates may change.

The coordinates change when window frame mode is used.

When the screen erase command (G202) is used, the erase color is as follows:

Color palette 15 when bit 6 (C9WN) of compile parameter No. 9100 is 1

Color palette 11 when bit 6 (C9WN) of compile parameter No. 9100 is 0

## N.6.4 Color Palette Difference between Character Display and Graphics Display

When display with a background color is specified with the CNC system, the color palette value for character display differs from the color palette value for graphics display in standard color specification. The color specified with the color type specification command (G240) differs between character display and graphics display.

# Index

## [A]



## [B]



## [C]



## [D]

# [E]



# [F]



# [G]



# [H]



# [I]

## [K]



## [L]



## [M]



## [N]



## [O]



## [P]



## [R]



## [S]

## [T]



## [U]



## [V]



## [W]

## Revision Record

**FANUC Series 16/18/20/21 PROGRAMMING MANUAL (Macro Compiler/Macro Executor) (B–61803E–1)**

| Edition | Date | Contents | Edition | Date | Contents |
|---|---|---|---|---|---|
| 05 | Dec., '94 | • Series 21 was added.<br>• Execution macro call mask function was added in 6.33. | | | |
| 04 | Mar., '94 | • Series 16–MODEL B was added.<br>• Graphic cursor function was added in 6.1.4.<br>• Rectangular display function was added in 6.1.5.<br>• Intensity modulation mode display of 9" monochrome CRT was added in 6.1.6. | | | |
| 03 | May, '93 | • Series 20 was added.<br>• Arithmetic function was added in 6.29.<br>• MDI key image read function was added in 6.30.<br>• Window function was added in 6.31. | | | |
| 02 | Oct., '92 | • Series 18 was added.<br>• All pages were revised. | 07 | Aug., '97 | • Series 16*i*/18*i*/21*i* was added. |
| 01 | May, '91 | ———— | 06 | Sep., '95 | • Series 21–MB was added. |

- *No part of this manual may be reproduced in any form.*
- *All specifications and designs are subject to change without notice.*