GE Fanuc Automation

Computer Numerical Control Products

DNC 2

Descriptions Manual

GFZ-61992E/02 *October 1997*

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.

Table of Contents

III. COMMUNICATION PROGRAM LIBRARY

I. GENERAL

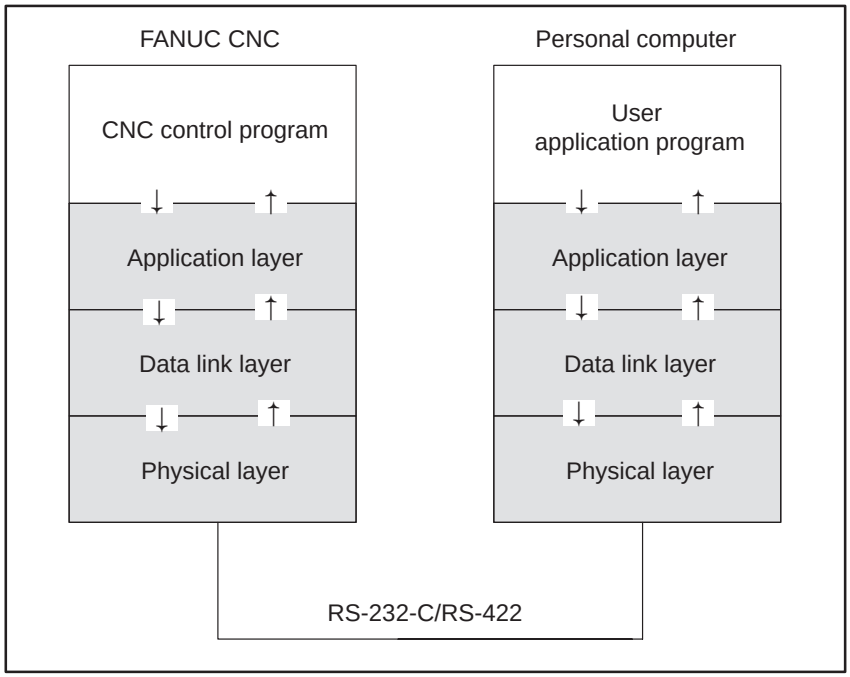
1 GENERAL

1.1 OUTLINE

The FANUC DNC2 is a communication protocol enabling data transmission between the FANUC CNC unit and a personal computer by connecting them via the RS-232-C or RS–422 interface.

The FANUC DNC2 has the following features:

(1) This protocol is based on the communication protocol LSV2 used by some CNC manufacturers in Europe, and has a three-layer structure so that software can easily be established even with a personal computer.
The RS-232-C interface is used to connect a personal computer with the FANUC CNC. The RS-422 interface can also be used to improve the transmission rate.



NOTE
The FANUC DNC2 provides compatibility with the LSV2 protocol for the physical and data link layers. The following table shows the compatibility of the application layer.

Table 1.1 Compatibility of Application Layer between DNC2 and LSV2

Service function	Compatibility
Service functions supported by both DNC2 and LSV2	Compatible
Service functions supported by DNC2 but not by LSV2	The DNC2 specifications have been extended.
Service functions supported by LSV2 but not by DNC2	DNC2 does not support these functions.

(2) This protocol is used for one-to-one (point-to-point) communication between one FANUC CNC unit and one personal computer.
The protocol cannot provide multi-point communication between one personal computer and more than one CNC unit.

(3) This protocol provides the various service functions listed in the following table. These functions enable automatic machining with data communication between the FANUC CNC and personal computer.

Service function		Service function	
Downloading a part program		Writing CNC data	CNC parameter
Uploading a part program			Pitch error compensation
Deleting a part program			Tool offset
Reading the directory of part programs			Custom macro variable
Reading information on a free area		Displaying an operator message	
Reading CNC data	Machine position	Reading PMC data	
	Absolute position	Writing PMC data	
	Skip position	Memory operation control	Selecting a program
	Servo delay		Start
	Acceleration/deceleration delay	DNC operation control	
	Machine interface signal status	Reset	
	Current program No.	Alarm notification	
	Current sequence No.	Report on change in the internal status of the CNC	
	CNC parameter	Providing information on CNC system identification	
	Pitch error compensation		
	Tool offset		
	Custom macro variable		
	Tool life management data		
	Modal data for automatic operation		
	Actual speed		
	A/D conversion data		
	Alarm information		
	Status information		

This manual applies to the following models:

Model	Abbreviation		
FANUC Series 0–MC	0–MC	Series 0–C	Series 0
FANUC Series 0–TC	0–TC		
FANUC Series 15–MA	15–MA	Series 15–A	Series 15
FANUC Series 15–TA	15–TA		
FANUC Series 15–TTA	15–TTA		
FANUC Series 15–MB	15–MB	Series 15–B	
FANUC Series 15–TB	15–TB		
FANUC Series 15–TTB	15–TTB		
FANUC Series 16–MA	16–MA	Series 16–A	Series 16
FANUC Series 16–TA	16–TA		
FANUC Series 16–GSA	16–GSA		
FANUC Series 16–GCA	16–GCA		
FANUC Series 16–MB	16–MB	Series 16–B	
FANUC Series 16–TB	16–TB		
FANUC Series 16–MC	16–MC	Series 16–C	
FANUC Series 16–TC	16–TC		
FANUC Series 16*i*–MA	16*i*–MA	Series 16*i*–A	
FANUC Series 16*i*–TA	16*i*–TA		
FANUC Series 18–MA	18–MA	Series 18–A	Series 18
FANUC Series 18–TA	18–TA		
FANUC Series 18–GSA	18–GSA		
FANUC Series 18–GCA	18–GCA		
FANUC Series 18–MB	18–MB	Series 18–B	
FANUC Series 18–TB	18–TB		
FANUC Series 18–MC	18–MC	Series 18–C	
FANUC Series 18–TC	18–TC		
FANUC Series 18*i*–MA	18*i*–MA	Series 18*i*–A	
FANUC Series 18*i*–TA	18*i*–TA		
FANUC Series 21*i*–MA	21*i*–MA	Series 21*i*–A	Series 21
FANUC Series 21*i*–TA	21*i*–TA		

For details of the functions supported by the Series 15–TTA/TTB, see Section II–4.2.2.

These functions cannot be used with the 2–path control of the Series 16/18.

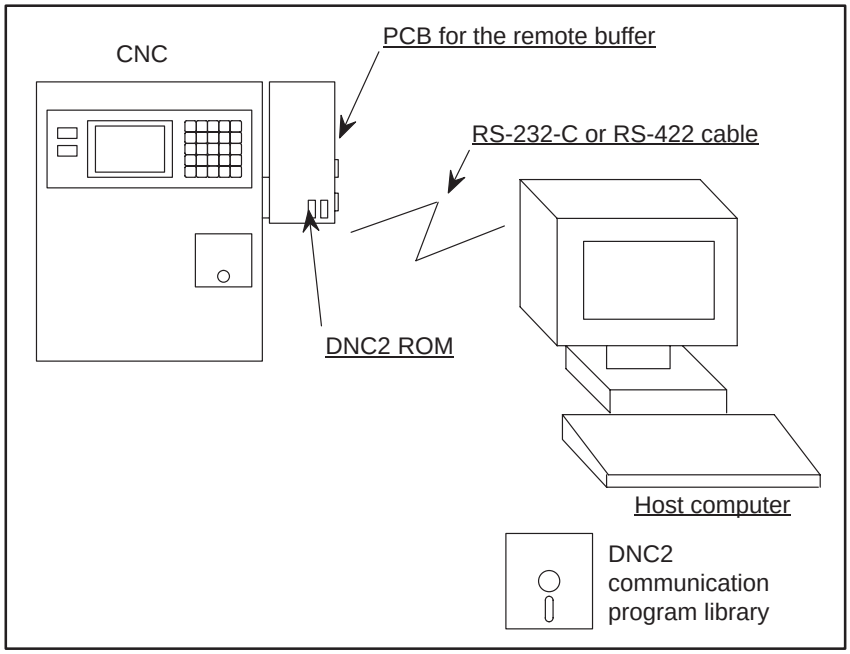
2 CONFIGURATION

The requirements for the DNC2 function are as follows:

- Hardware
  - DNC2 board
  - RS–232C/RS–422 cable
  - Host computer (supporting MS–DOS or PC–DOS)
- Host computer
  - DNC2 host software (A08B–9001–J530)
    * Library software for the host computer. Specify this software as necessary.
- CNC software options

Option	Series 0	Series 15	Series 16/18/21	Application
DNC2	○	○	○	DNC2 basic function
External data input/output	×	Δ	×	Operator message display
Stored pitch error compensation	Δ	Δ	Δ	Read/write of pitch error compensation
Custom macro (custom macro B for the Series 0/16/18)	Δ	Δ	Δ	Read/write of custom macro variables
Tool life management	Δ	Δ	Δ	Read of tool life management data
Background editing	○	×	○	DNC2 basic function
I/O unit external control	○	×	○	DNC2 basic function
PMC	Δ	–	–	Read/write of PMC data

○: Required
Δ: Optional
×: Not required
–: Standard

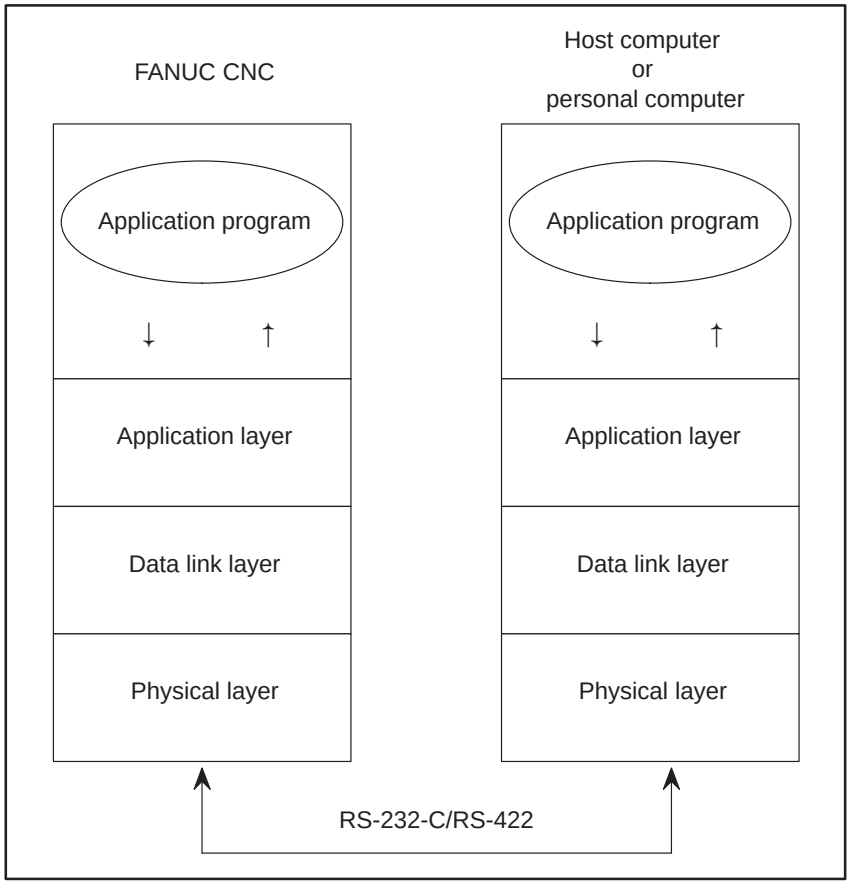


II. PROTOCOL

1 LAYERS

This protocol consists of the following three layers:

(1) Physical layer
(2) Data link layer
(3) Application layer



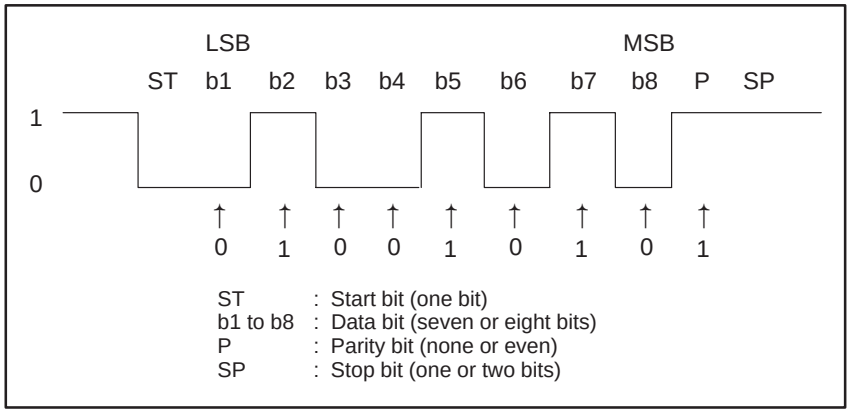
2 PHYSICAL LAYER

2.1 PHYSICAL INTERFACE

RS-232-C or RS-422 interface

2.2 SYNCHRONOUS METHOD

Asynchronous transmission



2.3 COMMUNICATION MODE

Half-duplex transmission

2.4 COMMUNICATION SPEED

(1) RS-232-C : 19200 bps max.

(2) RS-422 : 86400 bps max.
When the communication speed to be used is 38400 bps or more, synchronization with a receive clock is required. Prepare the TT (∗TT) and RT (∗RT) signals.

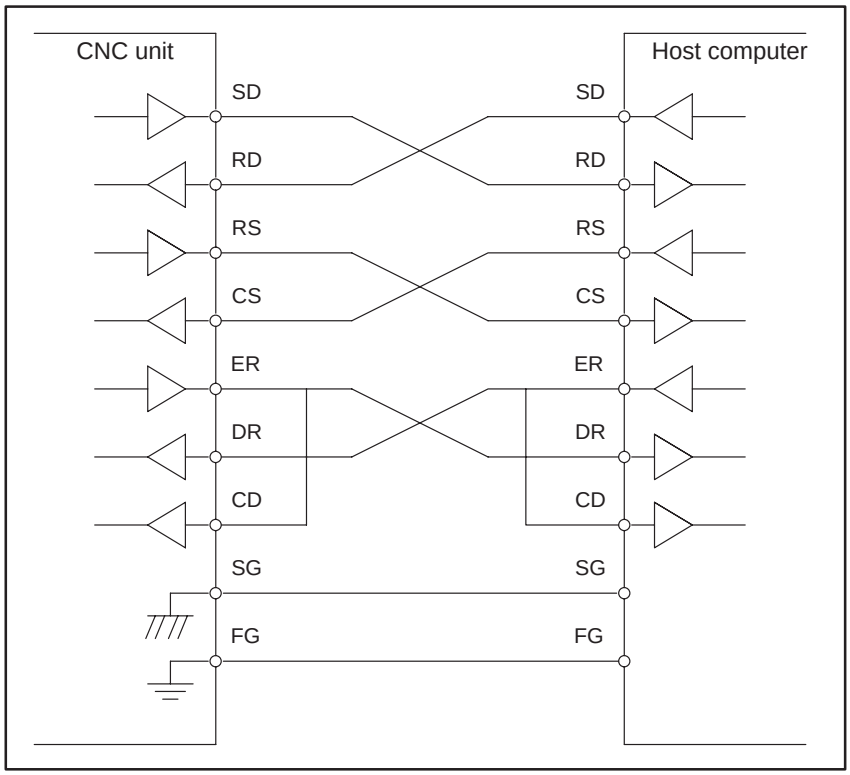
2.5 MAXIMUM CABLE LENGTH

(1) RS-232-C : 100m (for 4800 bps or less)
50m (9600 bps or more)

(2) RS-422 : 800m (9600 bps or less)
50m (19200 bps or more)

2.6 RS–232–C INTERFACE

(1) Connection between devices



When CS is not used, connect it with RS. When DR is not used, connect it with ER. Always connect CD with ER.

NOTE
When using IBM PC–AT, connect RS and CS in the CNC. (IBM PC–AT is a registered trademark of IBM Corporation.)

(2) Signals

Name	RS-232-C circuit No.	Input/ output	Description	
SD	103	Output	Send data	See Section 2.2 for the bit configuration.
RD	104	Input	Receive data	
RS	105	Output	Request to send This signal is used to report whether the CNC is ready to receive data. When the ER signal and this signal are on, the remote buffer is ready to receive data.	
CS	106	Input	Clear to send This signal is used to check whether the host computer is busy. When the DR signal and this signal are on, the host computer is assumed to be ready to receive data.	
DR	107	Input	Data set ready When this signal is on, the host computer is assumed to be ready. The signal is generally connected to the ER signal at the host computer. When it is off during data transmission, an alarm is issued. If this signal is not used, be sure to connect it with the ER signal at the CNC unit.	
ER	108.2	Output	CNC ready When this signal is on, the CNC is assumed to be ready to operate. The signal is generally connected to the DR signal at the host computer. When it is off during data transmission, an alarm is issued. If this signal is not used, be sure to connect it with the DR signal at the CNC unit.	
CD	109	Input	Signal quality detection Since this signal is not used for connecting the CNC unit with the host computer, connect it with the ER signal at the CNC unit.	
SG	102		Signal ground	
FG	101		Protective ground	

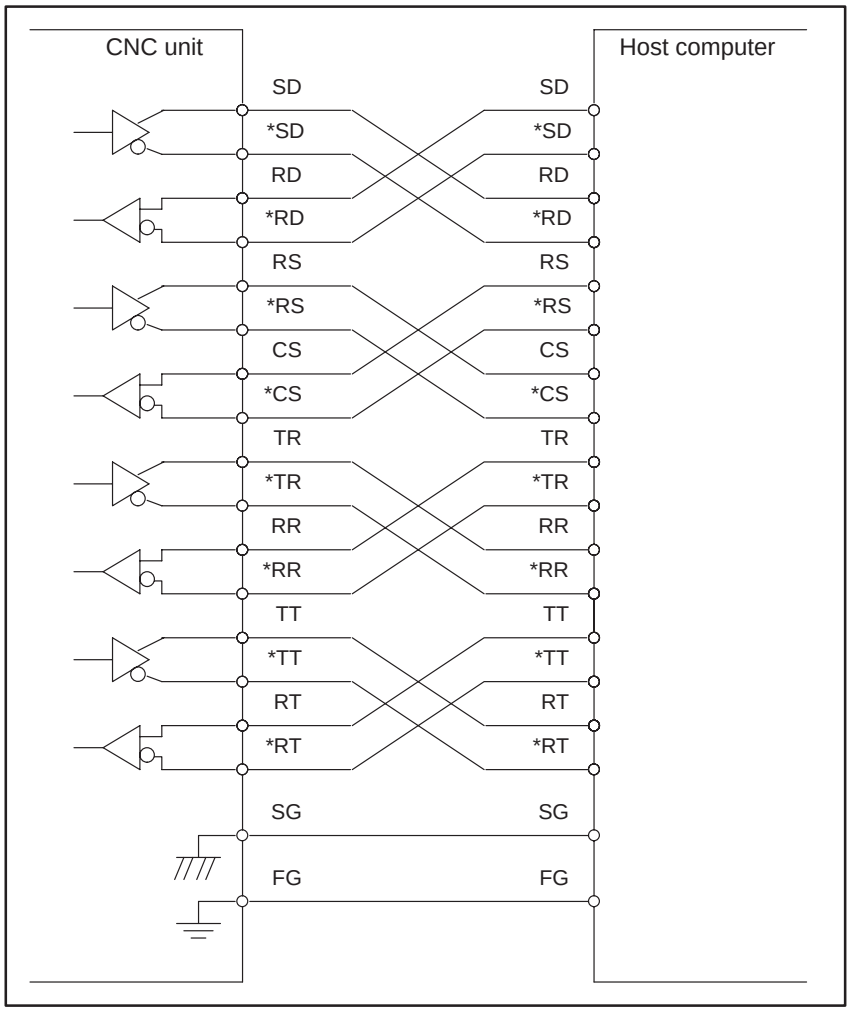
NOTE

The on and off states of each signal conform to the following:

	Less than –3V	+3V or more
Function	Off	On
Signal condition	Marking	Spacing

2.7 RS–422 INTERFACE

(1) Connection between devices



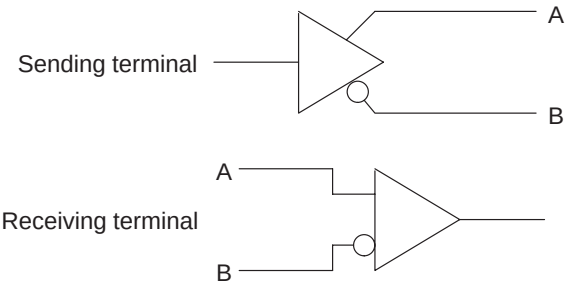
(2) Signals

Name	RS-422 circuit No.	Input/ output	Description	
SD	103	Output	Send data	See Section 3.2 for the bit configuration.
RD	104	Input	Receive data	
RS	105	Output	Request to send This signal is used to report whether the CNC is ready to receive data. When the TR signal and this signal are on, the remote buffer is ready to receive data.	
CS	106	Input	Clear to send This signal is used to check whether the host computer is busy. When the RR signal and this signal are on, the host computer is assumed to be ready to receive data.	
TR	108.2	Output	Terminal ready When this signal is on, the CNC is assumed to be ready to operate. The signal is generally connected to the RR signal at the host computer. When it is off during data transmission, an alarm is issued. If this signal is not used, be sure to connect it with the RR signal at the CNC unit.	
DM	109	Input	Receiver ready When this signal is on, the host computer is assumed to be ready. The signal generally indicates that the host computer is ready to send data to the CNC unit. If this signal is not used, be sure to connect it with the TR signal at the CNC unit.	
TT	113	Output	Transmission timing Send clock output terminal at the CNC unit. When a baud rate of 38400 or more is used, be sure to connect the terminal with the RT signal at the host computer.	
RT	115	Input	Reception timing Send clock input terminal at the CNC unit. When a baud rate of 38400 or more is used, be sure to connect the terminal with the TT signal at the host computer.	
SG	102		Signal ground	
FG	101		Protective ground	

NOTE

The on and off states of each signal conform to the following:

	A<B	A>B
Function	Off	On
Signal condition	Marking	Spacing



3 DATA LINK LAYER

The CNC unit and the host computer can operate on equal terms with each other under this protocol. If a contention occurs, however, the CNC is given priority over the host computer in data transmission.

3.1 TRANSMISSION CONTROL CHARACTERS

Table 3.1 lists the transmission control characters (TCCs).

Table 3.1 Transmission Control Characters

Tcc	Code (Hex)		Meaning
	ASCII	ISO	
ENQ	05H	05H	Request to send
NAK	15H	95H	Negative acknowledgment
DLE 0	10H 30H	90H 30H	Acknowledgment
DLE 1	10H 31H	90H B1H	Acknowledgment
DLE STX	10H 02H	90H 82H	Datagram start
DLE ETX	10H 03H	90H 03H	Datagram end
EOT	04H	84H	End of transmission

3.2 MESSAGE FORMAT

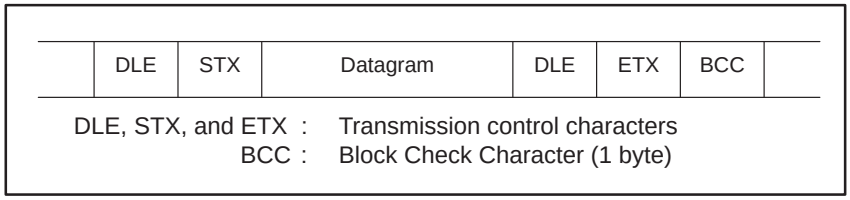


Fig.3.2 Message Format

(1) Datagram

The datagram consists of command and data sections.

The length of the command section is four characters.

The length of the data section is variable and can contain up to 256 characters.
The maximum length is fixed to 80 characters.

The datagram does not contain the communication control characters. The datagram must contain the command section. However, the data section can be omitted.

(2) Block check character (BCC)

The BCC is obtained by exclusive ORing all characters of the datagram excluding DLE and STX.

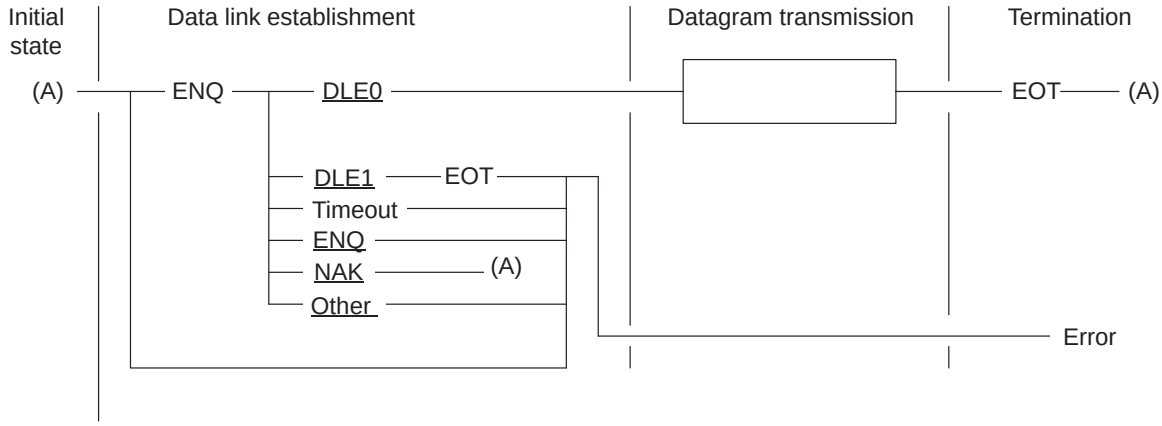
3.3 TRANSMISSION CHARACTERS

These characters are usually ASCII code but can also be ISO code.

3.4 DATA LINK PROTOCOL

3.4.1 Establishing a Data Link

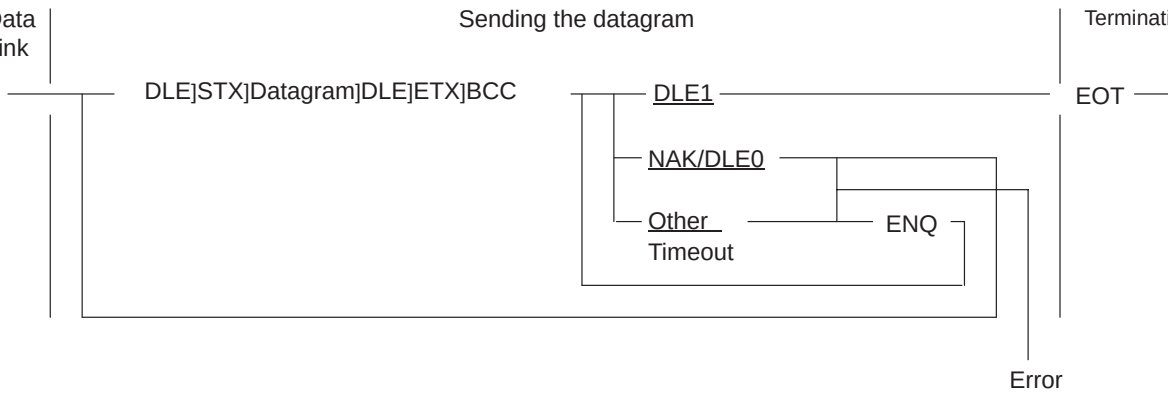
The master device that needs to send the datagram establishes the data link.



(Note) The underlined characters indicate the response from the receiving (slave) device.

Fig.3.4.1 Establishing the Data Link

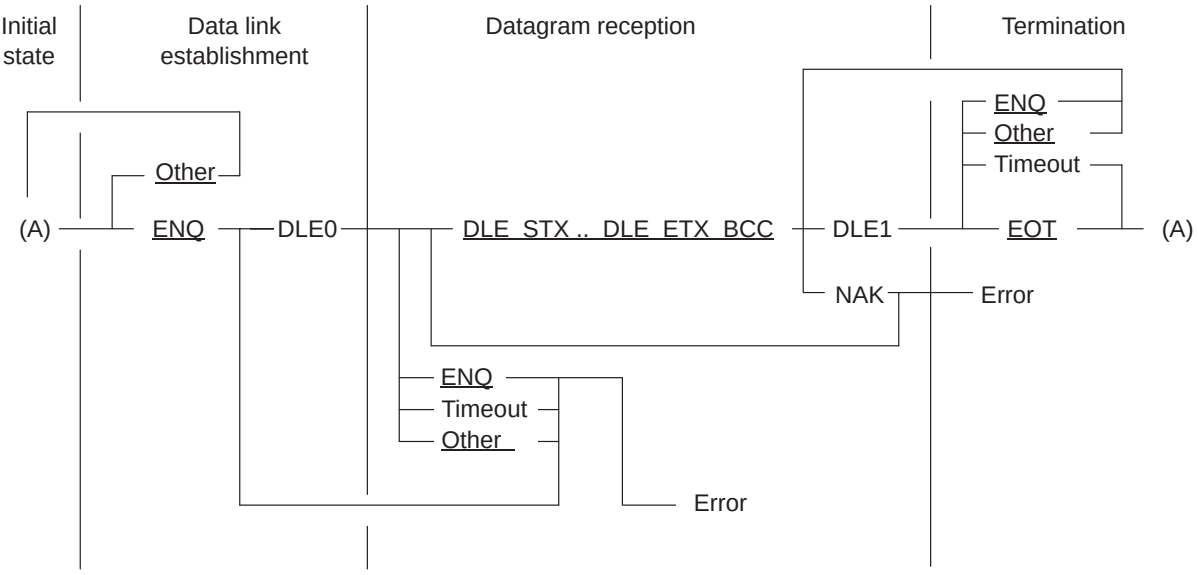
3.4.2 Sending a Datagram



(Note) The underlined characters indicate the response from the receiving (slave) device.

Fig.3.4.2 Sending a Datagram

3.4.3 Receiving a Datagram



(Note) The underlined characters indicate the response from the sending (master) device.

Fig.3.4.3 Receiving a Datagram

3.4.4 Termination

Datagram communication ends with transmission control character EOT.

4 APPLICATION LAYER

This chapter outlines the application layer. For more details, see the description given later.

4.1 FUNCTIONS

Table 4.1 lists the service functions.

Table 4.1 Service Functions

H: Host computer
C: CNC

Function		Direction of transmission	Description	Remarks
Downloading a part program		H↔C	Downloads a part program with the specified number from the host to the CNC.	
Uploading a part program		H↔C	Uploads a part program with the specified number from the CNC to the host.	
Deleting a part program		H→C	Deletes a part program with the specified number.	
Reading the directory information on a part program		H→C	Reads the program number registered in the tape storage size area.	
Reading CNC data	Machine position	H→C	Reads the machine position for the specified axis.	
	Absolute position	H→C	Reads the absolute position for the specified axis.	
	Skip position	H→C	Reads the skip position for the specified axis.	
	Servo delay	H→C	Reads servo delay for the specified axis.	
	Acceleration/deceleration delay	H→C	Reads acceleration/deceleration delay for the specified axis.	
	Diagnosis	H→C	Reads the states of the DI/DO signals with the specified numbers.	
	Current program No.	H→C	Reads the current program number.	
	Current sequence No.	H→C	Reads the current sequence number.	
	CNC parameter	H→C	Reads the value of the CNC parameter with the specified number.	
	Pitch error compensation data	H→C	Reads pitch error compensation data with the specified number.	
	Tool offset	H→C	Reads tool offset with the specified number.	
	Custom macro variable	H→C	Reads the custom macro variable with the specified number.	
	Tool life management data	H→C	Reads tool life management data with the specified tool group number.	
	Modal data for automatic operation	H→C	Reads the previous, current, or next modal data according to the specified parameters.	
	Actual speed	H→C	Read the actual feedrate.	
	A/D conversion data	H→C	Reads A/D data with the specified number.	
	Alarm information	H→C	Reads alarm information.	
	Status information	H→C	Reads status information.	
Writing CNC data	CNC parameter	H→C	Writes the specified data to the parameter with the specified number.	
	Pitch error compensation data	H→C	Writes the specified data to the pitch error compensation parameter with the specified number.	
	Tool offset	H→C	Writes the specified data to the tool offset parameter with the specified number.	
	Custom macro variable	H→C	Writes the specified data to the custom macro variable with the specified number.	
Operator message		H→C	Displays the specified message number and the message on the CRT.	
PMC data		H→C	Reads and writes PMC data with the specified address.	
Memory operation control		H→C	1) Selects a part program. 2) Starts a CNC cycle.	
DNC operation control		H→C	1) Selects a part program. 2) Starts downloading and starts a CNC cycle.	
Reset		H→C	Resets the CNC.	
Alarm notification		H←C	Reports the alarm in the CNC to the host.	
Status notification		H←C	Reports the internal status of the CNC to the host.	
System ID		H→C	Reads the model name and version of the CNC.	

4.2 DATAGRAM SYNTAX

The datagram consists of the command section and the omissible data section. The command section contains four characters. The data section contains variable-length data. The maximum length of the data section is specified by the CNC parameter. Subsections 4.2.1 and 4.2.2 list and explain the datagram commands. Subsection 4.2.4 describes the data section.

4.2.1 Configuration of Datagram Commands

Table 4.2.1 Configuration of datagram commands

	Fist entry										
A	Listing catalog	DI									
B	Catalog request	LI									
C	Clear	MC									
D	Operation mode	M									
E	Receive	R									
F	Transmit	T									
G	Prepare to send	PT									
H	Prepare to receive	PR									
	Second entry		A	B	C	D	E	F	G	H	
1	A/D Conversion data	AD			h/	/c					*
2	Acc./Dec error	AE			h/	/c					*
3	Actual feedrate	AF			h/	/c					*
4	Alarm status	AL			h/	/c					*
5	Abort (Break down)	BD			h/c						
6	Clear control (Reset)	CC					h/				
7	NC Start	CS					h/				
8	Operator message	DI					h/				
9	Command error	ER					h/c				
10	End of message	FD			h/c						
11	Free memory	FR			h/	/c					
12	System Identifier	ID			h/	/c					*
13	Modal data	MD			h/	/c					*
14	Machine I/F signal	MI			h/	/c					*
15	Machine position	MP			h/	/c					*
16	Macro variable	MV	h/	h/		h/c					*
17	Next block	NB			h/c						
18	Not position	NP			h/c		h/c				
19	Incorrect number	NR					h/c				
20	Function ready	OK					h/c				
21	Parameter	PA	h/	h/		h/c					*
22	PMC data	PD	h/	h/		h/c					*
23	Pitch error	PE	h/	h/		h/c					*
24	Part program	PM	h/c	h/c		h/c		h/	h/	/c	*
25	Part program number	PN			h/	/c					*
26	Ready to receive	RR					h/c				
27	Ready to transmit	RT					h/c				
28	Servo following error	SE			h/	/c					*
29	Select part program	SL					h/				*
30	Sequence number	SN			h/	/c					*
31	Skip position	SP			h/	/c					*
32	Status	ST			h/	/c	h/				
33	Tool life data	TL		h/		/c					*
34	Tool offset	TO	h/	h/		h/c					*
35	Absolute position	WP			h/	/c					*

NOTE

h/ : Command from the host to the CNC
/c : Command from the CNC to the host
Blank : Space character in ASCII or ISO code
* : Command modified or added by FANUC

4.2.2 Service Functions Specified with Datagram Commands

(1) Commands from the host

	Service function	Datagram command
1	Downloading	**<PR><PM><nn>**
2	Uploading	**<PT><PM><nn>**
3	Deleting a program	**<MC><PM><nn>**
4	Directory information	**<LI><PM> [<nn>]**
5	Free area size	**<T ><FR>**
6	Reading a tool position	**<T ><WP | MP | SP> [<0Xmm>]**
7	Reading servo delay	**<T ><SE> [<0Xmm>]**
8	Reading acceleration/deceleration delay	**<T ><AE> [<0Xmm>]**
9	Reading a machine interface signal	**<T ><MI><nn>[<:><mm>]**
10	Reading a program or sequence number	**<T ><PN | SN>**
11	Reading a CNC parameter	**<PT><PA><nn> [<,><0Xmm>]**
12	Writing a CNC parameter	**<PR><PA>**
13	Reading pitch error compensation	**<PT><PE><mm> [<:><nn>]**
14	Writing pitch error compensation	**<PR><PE>**
15	Reading a tool offset	**<PT><TO><mm> [<:><n>] <,><0Xkk>**
16	Writing a tool offset	**<PR><TO>**
17	Reading a macro variable	**<PT><MV><mm> [<:><nn>]**
18	Writing a macro variable	**<PR><MV>**
19	Reading tool life management data	**<PT><TL><mm> [<:><nn>]**
20	Reading modal information for automatic operation	**<T ><MD><mm>**
21	Reading an actual speed	**<T ><AF><0Xmm>**
22	Reading A/D conversion data	**<T ><AD><nn> [[<,><0Xmm>] <,><nn> [<,><0Xmm>]]**
23	Reading alarm information	**<T ><AL>**
24	Reading status information	**<T ><ST> | <M ><ST> [<0Xmm>]**
25	Displaying an operator message	**<M ><DI><nn><,><display message text>**
26	Reading PMC data	**<PT><PD><0Xaa | mm><,><nn>**
27	Writing PMC data	**<PR><PD>**
28	Selecting a program	**<M ><SL><nn>**
29	Executing a program	**<M ><CS> [<nn>]**
30	Resetting the CNC	**<M ><CC>**
31	Reading a system ID	**<T ><ID>**

NOTE
With the Series 15–TTA/TTB, nineteen functions are supported; none of these are related to the head. These functions are functions 1 to 9, 11 to 14, 22, 25 to 27, 30, and 31.

(2) Commands from the CNC

	Service function	Datagram command
1	Downloading	<PT><PM><nn>
2	Uploading	<PR><PM><nn>
3	Deleting a program	*
4	Directory information	*
5	Free area size	*
6	Reading a tool position	*
7	Reading servo delay	*
8	Reading acceleration/deceleration delay	*
9	Reading machine interface signals	*
10	Reading a program or sequence number	*
11	Reading a CNC parameter	*
12	Writing a CNC parameter	*
13	Reading pitch error compensation	*
14	Writing pitch error compensation	*
15	Reading a tool offset	*
16	Writing a tool offset	*
17	Reading a macro variable	*
18	Writing a macro variable	*
19	Reading tool life management data	*
20	Reading modal information for automatic operation	*
21	Reading an actual speed	*
22	Reading A/D conversion data	*
23	Reading alarm information	*
24	Reading status information	<R ><ST><0Xss> | <R ><AL><0Xaa>
25	Displaying an operator message	*
26	Reading PMC data	*
27	Writing PMC data	*
28	Selecting a program	*
29	Executing a program	*
30	Resetting the CNC	*
31	Reading a system ID	*

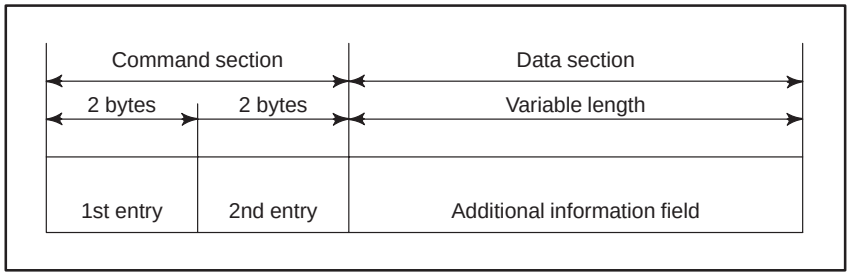
NOTE
The service functions marked with an asterisk (*) are not supported.

4.2.3 Notation of the Datagram

4.2.3.1 Symbols

< > : Datagram entry
[ ] : Brackets for a datagram entry (A datagram entry enclosed in these brackets can be omitted.)
{ } : Braces for a datagram entry (A datagram entry enclosed in these braces must not be omitted.)
| : Delimiter for datagram entries
0X : Prefix of hexadecimal data
Blank : Space character in ASCII or ISO code

4.2.3.2 Format



The length of the data section is variable. The maximum length is specified by the CNC parameter. The parameter can specify the length from 80 to 256 characters.

4.2.3.3 Data type

The data type used in Subsection 4.2.4 is shown below.

(1) First entry
UI : Unsigned integer
SI : Signed integer
CH : Character
UR : Unsigned real number
SR : Signed real number
HX : Hexadecimal number

Hexadecimal data must begin with 0X.
Data of UI and UR types can contain no sign. The positive sign of a positive value is omitted.

(2) Second entry

The second entry indicates the maximum number of digits or characters for each data type.

The digits and characters do not contain a sign or a decimal point, however.

Example)

	(Correct)	(Incorrect)
SI:4 –	–9999 to 9999	12345
SR:9 –	–99999999. to 99999999.	123.4567895
HX:4 –	0X19A5	0XFF1234
CH:8 –	ABCDEFGH or ABC EF	ABCDEFGHI

(3) Third entry

When the value is clearly determined, this entry indicates the range and limit values.

Example)

UI : 2/1 to 15
Unsigned integer : First entry
Maximum of two digits : Second entry
Range from 1 to 15 : Third entry

4.2.3.4 Delimiter in the data section

Only a comma (,) and colon (:) can be used as a delimiter. A space is ignored except when it is used for character–type data.

The end of the data section is assumed to be a delimiter.

4.2.3.5 Code

ASCII or ISO code is available. The code to be used in the datagram is specified by the CNC parameter.

4.2.3.6 Continuation

The data section can be continued only for the datagram of the response to the <PR> or <PT> command.

If the length of the data section exceeds the maximum length, the data is divided into more than one datagram when it is transmitted.

Continuation in the data section is not limited except for group data that consists of multiple data items. (The group data cannot be divided when it is transmitted.)

Example)

Correct)

```
<R ><PM><G01X100.Y–1>   →  "G01X100.Y–123.4Z50.0F200;"
<R ><PM><23.4Z50.0F200;>
```

Incorrect)

```
<R ><TO><N10D15H5N11D16>→  "N11D16H6" cannot be divided when it
<R ><TO><H6N12D17H7>       is transmitted.
                             "N– D– H–" is a group data item.

<R ><PD><N0X100P0XAAN0X101> →"N– P–" is a group data item.
<R ><PD><P0X55>
```

4.2.4 Details of the Data Section

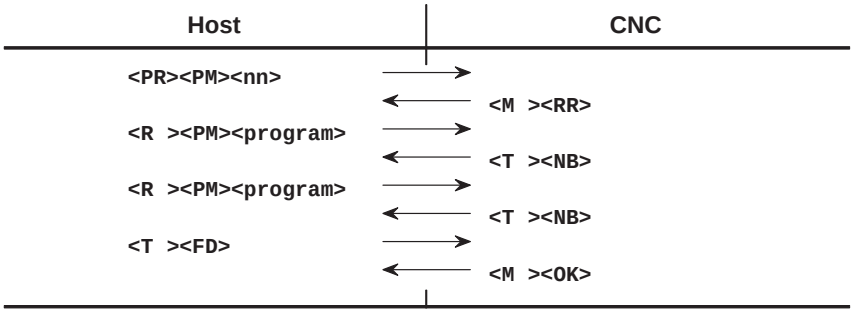
4.2.4.1 Downloading a part program

(1) Function

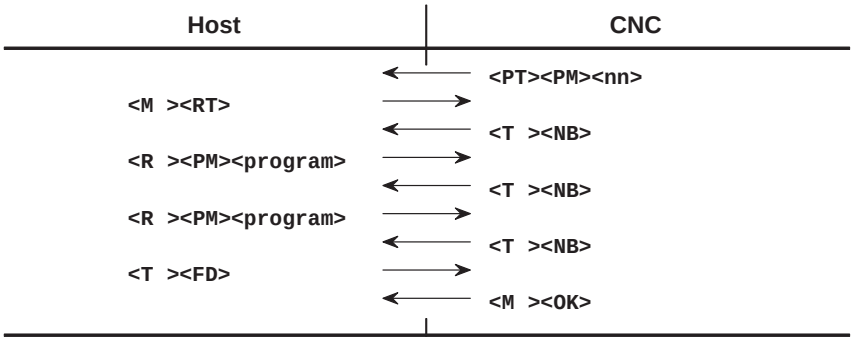
1) This command transfers a part program to the CNC according to the commands issued by the host computer.
2) This command transfers a part program to the CNC according to the commands issued by the CNC.

(2) Datagram transmission

1) Commands issued by the host computer



2) Commands issued by the CNC



(3) Parameters

1) nn : Program number (UI: 4)
Only a program can be uploaded using this parameter.
2) program: Program statement (CH: 256)
This statement is in the tape output format.
The program consists of the following:

NOTE
The above program number nn must be the same as nn in command <PT | PR> <PM> <nn>. The semicolon (;) functions as end of block. Actually, the semicolon is a EOB code defined in the CNC such as an LF code (0AH).

(4) Negative acknowledgment

1) **M_NR [0XF61F]**:
The program with the same number has already been found.

2) **M_NP [0XF62D]**:
The program is write–protected.

3) **M_NP [0XFDFB]**:
An alarm is raised in the CNC. (For Series 0/16/18/21)

4) **T_BD [0XF61E]**:
Insufficient free area in memory

5) Note
Release background edit mode and simultaneous edit mode (Series 15 only) in advance.

4.2.4.2 Uploading a part program

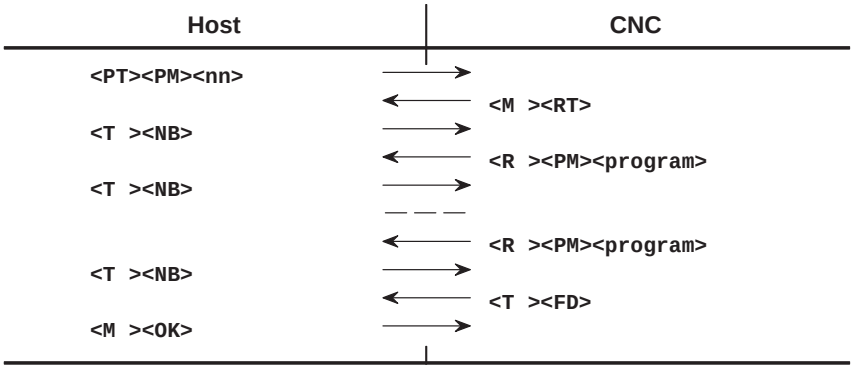
(1) Function

1) This command transfers a part program from the CNC according to the commands issued by the host computer.

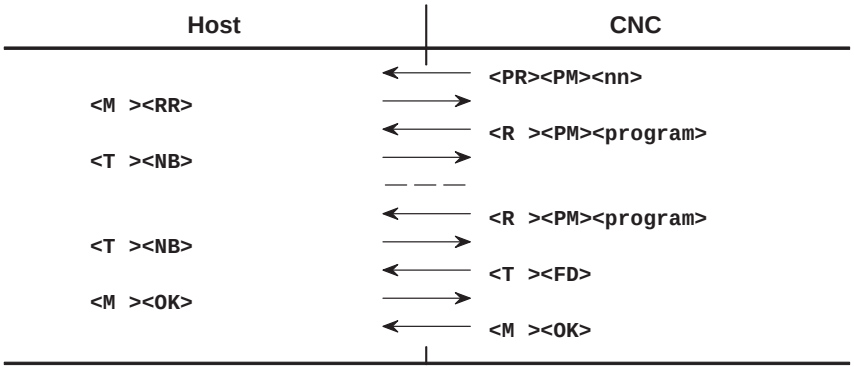
2) This command transfers a part program from the CNC according to the commands issued by the CNC.

(2) Datagram transmission

1) Commands issued by the host computer



2) Commands issued by the CNC



(3) Parameters

1) nn : Program number (UI: 4)
Only a program can be downloaded using this parameter.

2) program: Program statement (CH: 256)
This statement is in the tape output format.
See 2) in (3) of Subsection 4.2.4.1.

(4) Negative acknowledgment

1) **M_NR [0XF625]** : No program is found.
2) **M_NP [0XFDFB]** : An alarm occurred in the CNC. (For Series 0 only)

(5) Note

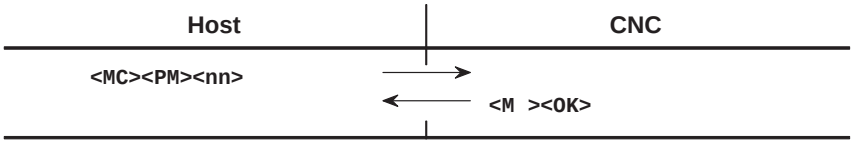
Release background edit mode and simultaneous edit mode (Series 15 only) in advance.

4.2.4.3 Deleting a part program

(1) Function

This command deletes a part program from memory.

(2) Datagram transmission



(3) Parameter

nn : Program number (SI: 4)
If nn is –9999, all programs are deleted.

(4) Negative acknowledgment

1) **M_NR [0XFB9D]**:
The program with nn is not found, or no programs are found.
2) **M_NP [0XFB98]**:
The program is protected by a parameter or key switch.
3) **M_NP [0XFB93]**:
The program is being used (selected or executed). If the program is being selected in the edit mode, it can be deleted.
4) **M_NP [0XFBA7]**:
Alarm state (for Series 0)
nn is no –9999, and the system is not in the edit mode.
(for Series 15/16/18/21)

(5) Notes

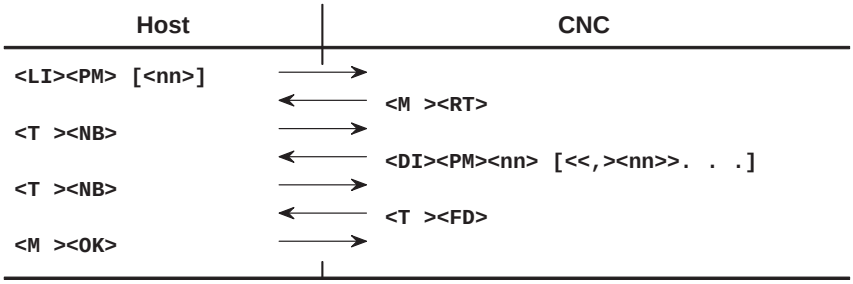
1) In Series 15, the currently selected program cannot be deleted if the command for deleting all programs is issued in the memory mode.
2) Release background edit mode and simultaneous edit mode (Series 15 only) in advance.

4.2.4.4 Reading the directory of part programs

(1) Function

This command reads the directory of part programs in memory.

(2) Datagram transmission



(3) Parameter

nn: Program number (UI: 4)
If the program number is omitted, all program numbers are reported.
If the program number is specified, the host computer checks whether the program is found in program memory.

(4) Negative acknowledgment

1) **T_NP [0XFC02]**:
The specified program is not found. (For Series 0/16/18/21)

2) **T_NP [0XFC03]**:
The specified program is not found. (For Series 15)

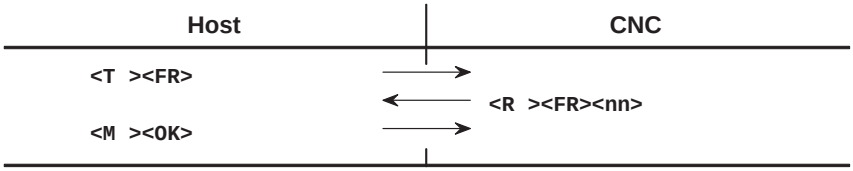
3) **T_NP [0XFC02]**:
No program is found. (When no program)

4.2.4.5 Free area in program memory

(1) Function

This command issues a request from the host computer to the CNC for sending data on the size of the free area in program memory in bytes.

(2) Datagram transmission



(3) Parameter

nn : Size of the free area in program memory (UI: 9)

(4) Negative acknowledgment

None

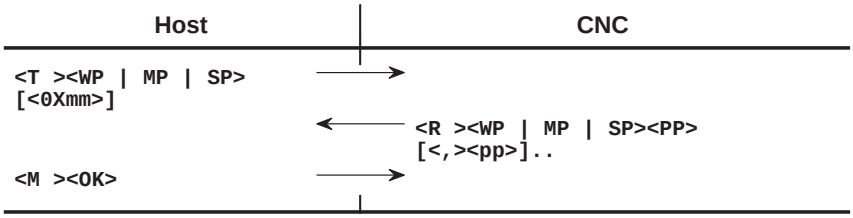
4.2.4.6 Reading a tool position

(1) Function

This command reads the current tool position.

1) Absolute coordinate position
2) Machine coordinate position
3) Skip position

(2) Datagram transmission

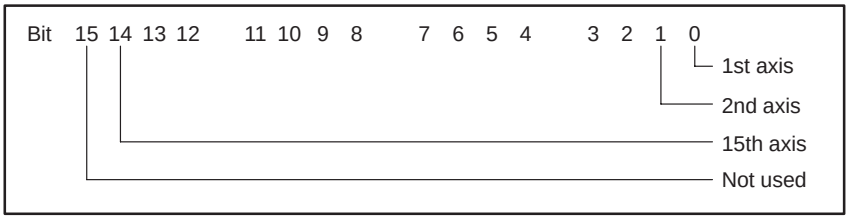


NOTE

<T><WP> : Absolute coordinate position
<T><MP> : Machine coordinate position
<T><SP> : Skip signal detection position

(3) Parameters

1) mm: Axis request flag (HX: 4)
If mm is omitted, all axis positions are transferred.
0X indicates that mm must be specified in hexadecimal.



If 0X3F is specified, the 1st to 6th axis positions are transferred.

2) pp : Tool position data (SR: 9)
   a) pp is repeated in ascending order of axis numbers.
   b) The number in pp varies according to the axis request flag.

(4) Negative acknowledgment

M_NR [0XFFCC]: No specified axis is found.

(5) Note

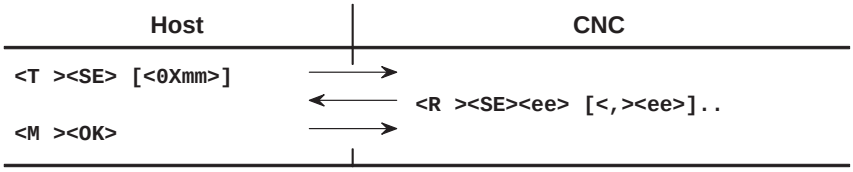
When the tool position read function is being used with the Series 15, bit 1 (POS) of parameter 2204 must be set to 1.

With the Series 15, when absolute coordinates and machine coordinates are to be read, POS of parameter 2204 must be set to 1.

4.2.4.7 Servo delay

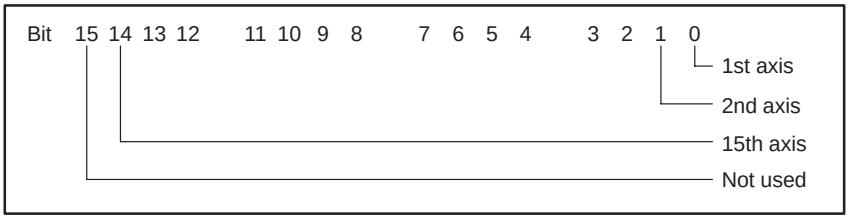
(1) Function
This command reads the current servo delay.

(2) Datagram transmission



(3) Parameters

1) mm: Axis request flag (HX: 4)
If mm is omitted, all axis positions are transferred.
0X indicates that mm must be specified in hexadecimal.



If 0X3F is specified, the 1st to 6th axis positions are transferred.

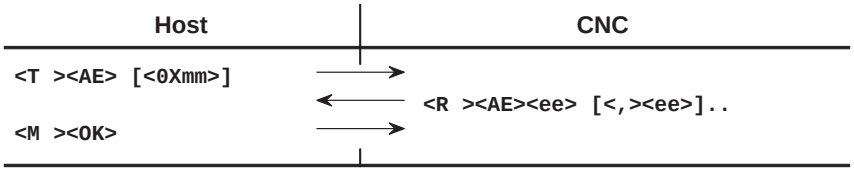
2) ee : Tool position data (SR: 9)
   a) ee is repeated in ascending order of axis numbers.
   b) The number in ee varies according to the axis request flag.

(4) Negative acknowledgment
M_NR [0XFFCC]: No specified axis is found.

4.2.4.8 Acceleration/ deceleration delay

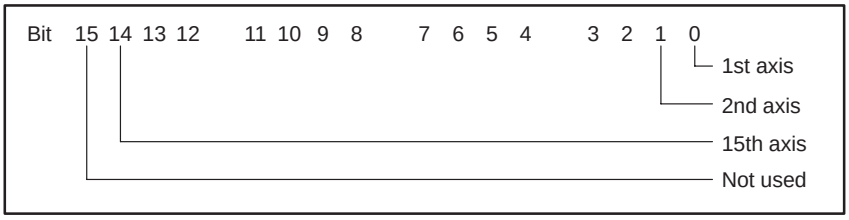
(1) Function
This command reads the current acceleration/deceleration delay.

(2) Datagram transmission



(3) Parameters

1) mm: Axis request flag (HX: 4)
If mm is omitted, all axis positions are transferred.
0X indicates that mm must be specified in hexadecimal.



If 0X3F is specified, the 1st to 6th axis positions are transferred.

2) ee : Current acceleration/deceleration delay (SR: 9)
   a) ee is repeated in ascending order of axis numbers.
   b) The number in ee varies according to the axis request flag.

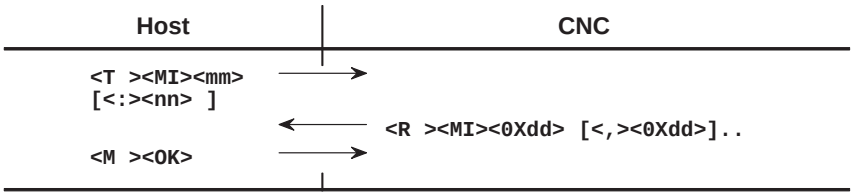
(4) Negative acknowledgment
M_NR [0XFFCC]: No specified axis is found.

4.2.4.9 Machine interface signals (For Series 15 only)

(1) Function

This command reads the machine interface signals (DI/DO) specified with diagnosis numbers.

(2) Datagram transmission



(3) Parameters

1) mm: First diagnosis number to be read (UI: 3)
2) nn : Last diagnosis number to be read (UI: 3/mm < nn)
The DI/DO signal data items from mm to nn are transmitted.
If nn is omitted, one signal data item is transmitted.
3) dd : DI/DO signal data (HX: 2)
The data is transmitted in hexadecimal.

(4) Negative acknowledgment

M_NR [0XFFCA]: The diagnosis number is incorrect.

(5) Note

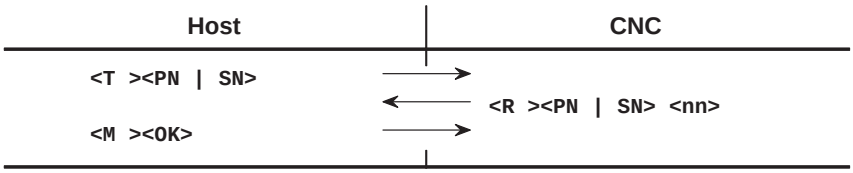
The maximum number of DI/DO data items are limited to L/5 by the length of the datagram.
(L is the length of the datagram.)

4.2.4.10 Current program number and sequence number

(1) Function

This command reads the current program number and sequence number.

(2) Datagram transmission



NOTE
<M><PN>: Reads the program number.
<M><SN>: Reads the sequence number.

(3) Parameter

nn : Current program number (UI: 4) or the latest sequence number specified in the program (UI: 4 for Series 0/16/18/21 or UI: 5 for Series 15)

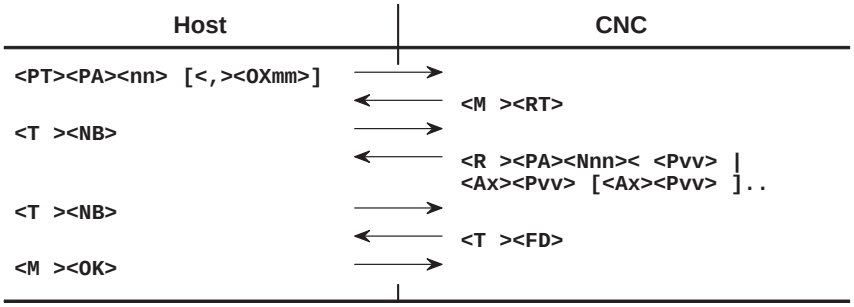
(4) Negative acknowledgment

None

4.2.4.11 Reading a CNC parameter

(1) Function
This command reads a CNC parameter.

(2) Datagram transmission



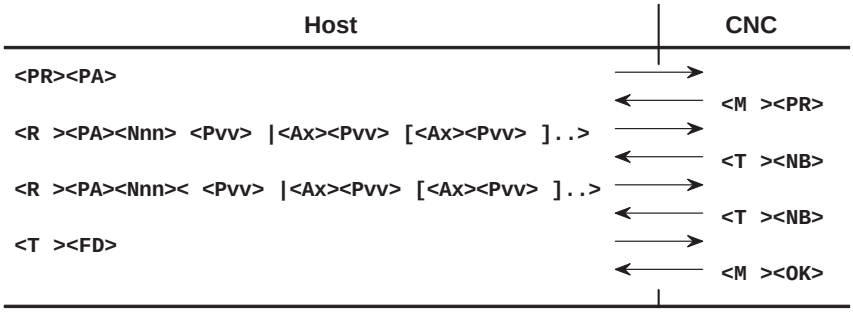
(3) Parameters
1) nn : Number of the parameter to be read (UI: 4)
2) mm : Axis request flag (HX: 4)
mm must be specified when the parameter is of axis type.
3) Nnn : nn is the parameter number. (UI:4)
4) Pvv : vv is the parameter value. (SI:10)
5) Ax : x is the axis exponent. (UI: 2/1 to 15)
If nn is of axis type, Ax is transmitted.

(4) Negative acknowledgment
1) **M_NR [0XFBA5]**:
The parameter number is incorrect.(For Series 0 only)
2) **M_NR [0XFBA4]**:
The parameter number is incorrect.(For Series 15 only)
3) **M_NR [0XFBA3]**:
The axis exponent parameter (0Xmm) is invalid.
(For Series 15 only)

4.2.4.12 Writing a CNC parameter

(1) Function
This command writes a CNC parameter.

(2) Datagram transmission



(3) Parameters
1) Nnn : nn is the parameter number. (UI: 4)
2) Pvv : nn is the parameter value. (SI: 10)
3) Ax : x is the axis exponent. (UI: 2/1 to 15)
If nn is of axis type, Ax is transmitted.

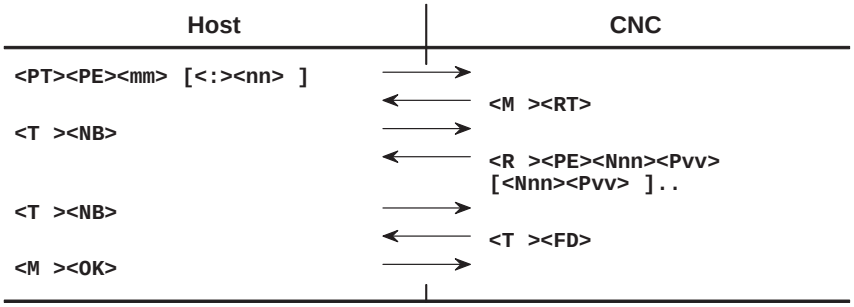
(4) Negative acknowledgment
1) **M_NP [0XFB98]**:
The parameter is write–protected or locked.
(For Series 15/16/18/21)
2) **M_NR [0XFBA45]**:
The parameter is invalid. (For Series 0/16/18/21)
3) **M_NR [0XFBA4]**:
The parameter is invalid. (For Series 15/16/18/21)

4.2.4.13 Reading pitch error compensation data

(1) Function

This command reads pitch error compensation data.

(2) Datagram transmission



(3) Parameters

1) mm : First pitch error compensation number to be read (UI: 4) With the Series 15/16/18/21, the number starts with 0. With the Series 0, the number starts with 1000. (This number is the same as the parameter number.)

2) nn : Last pitch error compensation number to be read (UI: 4 / $1 \leqq mm < nn \leqq 9999$)

3) Nnn : nn is the pitch error compensation number. (UI: 4)

4) Pvv : vv is the compensation corresponding to the specified pitch error compensation number. (SI: 1 / $-7 \leqq vv \leqq 7$)

(4) Negative acknowledgment

M_NR [0XFFCA]:

The parameter value is invalid, or no option is found (for Series 0 only).

(5) Note

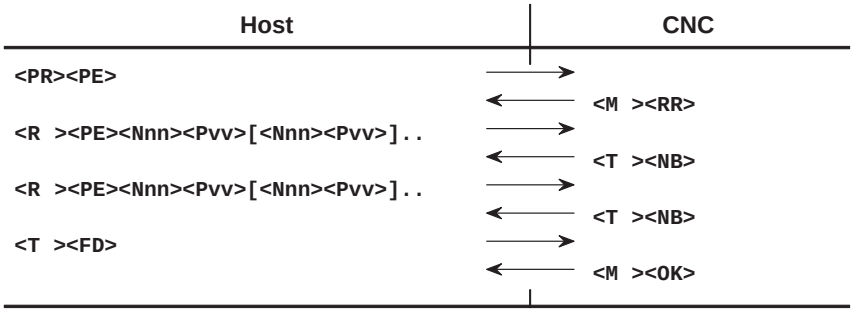
This command can read only the pitch error compensation from the CNC. To read other parameters, use the <PT><PA> command.

4.2.4.14 Writing pitch error compensation

(1) Function

This command writes pitch error compensation.

(2) Datagram transmission



(3) Parameters

1) Nnn : nn is the pitch error compensation number. (UI: 4)
2) Pvv : Compensation corresponding to the specified pitch error compensation number (SI: 1 / –7 ≦ vv ≦ 7)

(4) Negative acknowledgment

1) **M_NR [0XFBA5]**:
   The parameter is invalid. (For Series 0/16/18/21)
2) **M_NR [0XFBA4]**:
   The parameter is invalid. (For Series 15)
3) **M_NP [0XFB9E]**:
   No option is found. (For Series 0/16/18/21)

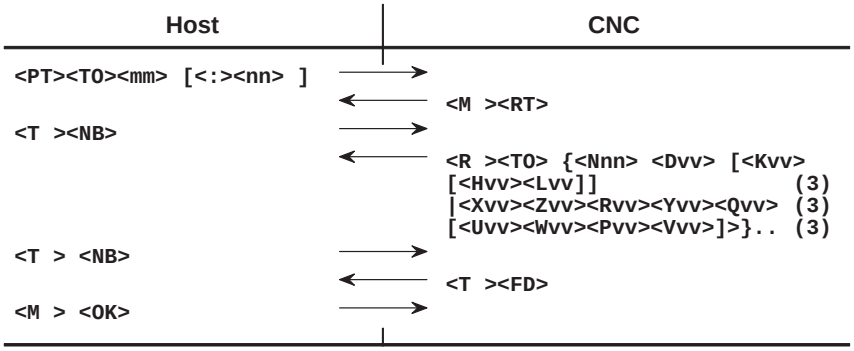
(5) Note

This command can write only the pitch error compensation into the CNC. To write other parameters, use the <PR> <PA> command.

4.2.4.15 Reading a tool offset

(1) Function
This command reads a tool offset.

(2) Datagram transmission



(3) Parameters
1) mm: First tool offset number to be read (UI: 4)
2) nn : Last tool offset number to be read
(UI: 4 / 1 ≦ mm <nn ≦ 9999)
3) kk : Offset request flag (HX: 4)

bit

0	:Wear offset	D :	Cutter compensation
1	:Geometry offset	K :	Cutter compensation
2	:Wear offset	X :	Offset along the X axis
3	:Wear offset	Z :	Offset along the Z axis
4	:Wear offset	R :	Tool tip radius compensation
5	:Wear offset	Y :	Offset along the Y axis
6	:Virtual tool tip direction	:	Q
7	:Not used		
8	:Wear offset	H :	Tool length offset
9	:Geometry offset	L :	Tool length offset
10	:Geometry offset	U :	Offset along the X axis
11	:Geometry offset	W :	Offset along the Z axis
12	:Geometry offset	P :	Tool tip radius compensation
13	:Geometry offset	V :	Offset along the Y axis
14	:Not used		
15	:Not used		

NOTE
(D, K, H, L) : Offsets for the milling machine
(X, Z, R, Y, Q, U, W, P, V) : Offsets for the lathe
Both the offset request flags must be mutually exclusive when they are set.
Example) 0X3F00 : Incorrect
0X3C7C : Correct

4) Nnn : Tool offset number (UI: 4)

5) Dvv : Cutter wear offset for the milling machine (SR: 9)
Kvv : Cutter geometry offset for the milling machine (SR: 9)
Hvv : Tool length wear offset for the milling machine (SR: 9)
Lvv : Tool length geometry offset for the milling machine (SR:9)
Xvv : Wear offset along the X axis for the lathe (SR: 9)
Zvv : Wear offset along the Z axis for the lathe (SR: 9)
Rvv : Tool tip radius wear offset for the lathe (SR: 9)
Yvv : Wear offset along the Y axis for the lathe (SR:9)
Qvv : Virtual tool tip direction for the lathe (UI: 1 / 0 ≦ vv ≦ 7)
Uvv : Geometry offset along the X axis for the lathe (SR: 9)
Wvv : Geometry offset along the Z axis for the lathe (SR: 9)
Pvv : Tool tip radius geometry offset for the lathe (SR: 9)
Vvv : Geometry offset along the Y axis for the lathe (SR: 9)

(4) Negative acknowledgment

1) **M_NR [0XFFCA]**:
The offset number is incorrect. (mm, nn, or 0Xkk)

2) **M_NR [0XFFC9]**:
Combination of offsets in incorrect.

(5) Note

Table 4.2.4.15 describes how the tool compensation memory type and request flag are specified.

Table 4.2.4.15 Relationships between Tool Compensation Memory Type and Request Flag

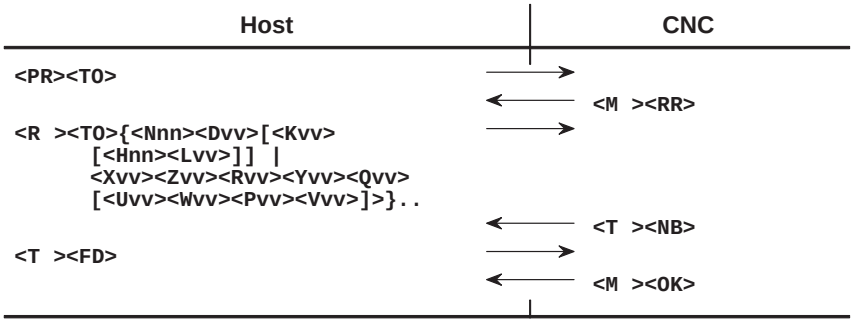
Model	Memory type		Request flag 15	14	12	12	11	10	9	8	7	6	5	4	3	2	1	0
			–	–	V	P	W	U	L	H	–	Q	Y	R	Z	X	K	D
			–	–	T				M			T					M	
Series0	T	b	–	–	–	–	–	–	×	×	–	○	○	○	○	○	×	×
		o	–	–	×	×	×	×	×	×	–	○	○	○	○	○	×	×
	M	b	–	–	×	×	×	×	–	–	–	×	×	×	×	×	–	○
		o	–	–	×	×	×	×	–	–	–	×	×	×	×	×	–	○
Series15	T	A	–	–	–	–	–	–	×	×	–	○	○	○	○	○	×	×
		B	–	–	○	○	○	○	×	×	–	○	○	○	○	○	×	×
	M	A	–	–	×	×	×	×	×	×	–	×	×	×	×	×	×	○
		B	–	–	×	×	×	×	×	×	–	×	×	×	×	×	○	○
		C	–	–	×	×	×	×	○	○	–	×	×	×	×	×	○	○
Series 16/18/21	T	b	–	–	–	–	–	–	×	×	–	○	○	○	○	○	×	×
		o	–	–	○	○	○	○	×	×	–	○	○	○	○	○	×	×
	M	b	–	–	×	×	×	×	×	×	–	×	×	×	×	×	○	○
		o	–	–	×	×	×	×	○	○	–	×	×	×	×	×	○	○

– : Ignored b : Standard
○ : Specifiable o : Option
× : Not specifiable

4.2.4.16 Writing a tool offset

(1) Function
This command writes a tool offset.

(2) Datagram transmission



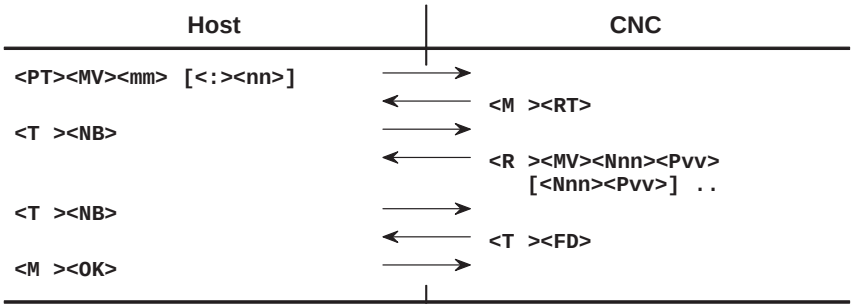
(3) Parameters
See 3) and 4) in item (3) of Subsection 4.2.4.15.

(4) Negative acknowledgment
1) **M_NR [0XFBA3]** :
The compensation number is incorrect. (For Series 0 only)
2) **M_NR [0XFBA4]** :
The compensation number is incorrect. (For Series 15 only)

4.2.4.17 Reading custom macro variables

(1) Function
This command reads custom macro variables.

(2) Datagram transmission



(3) Parameters
1) mm : First custom macro variable number to be read (UI:5)
2) nn : Last custom macro variable number to be read (UI:5 / mm < nn)
3) Nnn : nn is the custom macro variable number. (UI:5)
4) Pvv : Value of custom macro variable nn (SR:9)

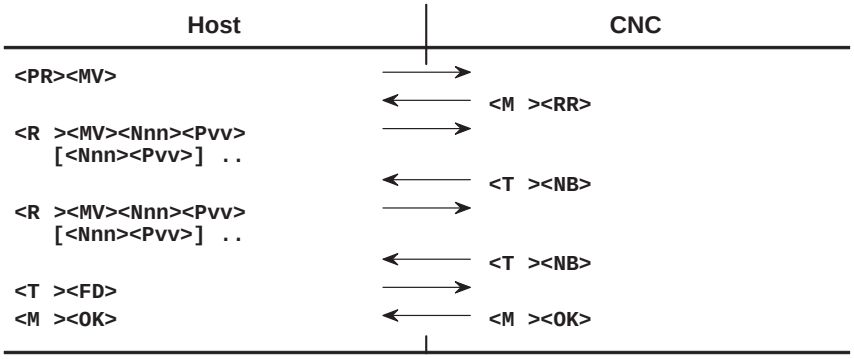
(4) Negative acknowledgment
M_NR[0XFFCA] :
The custom macro variable number is incorrect, or no option is found.

4.2.4.18 Writing a custom macro variable

(1) Function

This command writes a value in the specified custom macro variable.

(2) Datagram transmission



(3) Parameters

1) Nnn : nn is the custom macro variable number. (UI:5)
2) Pvv : Value of custom macro variable nn (SR:9)

(4) Negative acknowledgment

1) **M_NR [0XFBA5]** :
   The custom macro variable number is incorrect.
   (For Series 0/16/18/21)
2) **M_NR [0XFFA4]** :
   The custom macro variable number is incorrect.
   (For Series 15)
3) **M_NR [0XFB9E]** :
   No option is found.

(5) Note

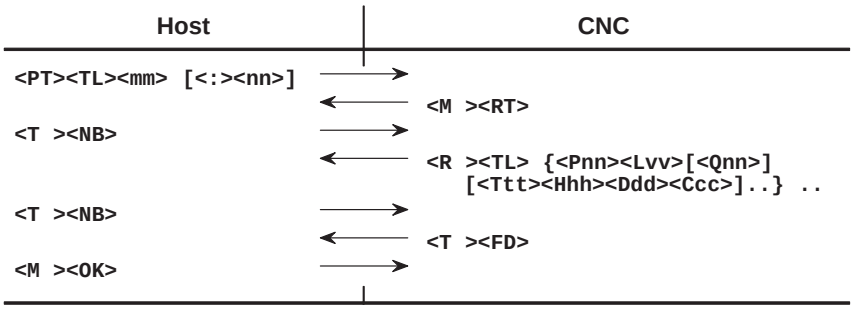
With the Series 0, the custom macro variables in which data can be written are #100 to #149 and #500 to #531 of custom macro A (integer data). Data cannot be written into the extended area.

4.2.4.19 Reading tool life management data

(1) Function

This command reads tool life management data with the specified group number.

(2) Datagram transmission



(3) Parameters

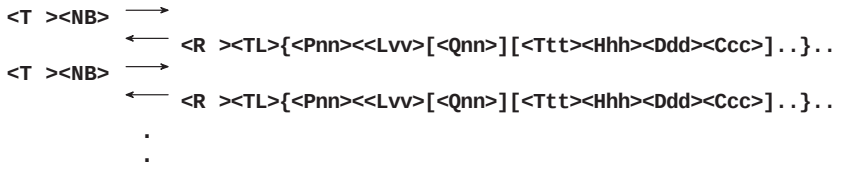
1) mm : First tool group number to be read (UI:4)
2) nn : Last tool group number to be read (UI:4 / mm < nn)
3) Pnn : nn is the tool group number. (UI:4)
4) Lnn : Tool life value (minutes or times) (UI:10)
5) Qnn : Qnn: nn is the life count value of the tool currently being used. (UI: 5) The unit is the number of times the tool has been used or the total usage time (in minutes), one of which is specified with a parameter of the CNC.
6) Ttt : Tool number (UI:4)
7) Hhh : H code (UI:4)
8) Ddd : D code (UI:4)
9) Ccc : Tool information (UI:1/0 to 3)
   0 : Currently selected tool
   1 : The tool is already registered.
   2 : The tool is already exhausted.
   3 : The tool was skipped.

(4) Negative acknowledgment

M_NR [0XFFCA] :
Group number nn is not yet defined, or no option is found.

(5) Note

If the tool life management data cannot be transmitted with one datagram, the datagram is divided into multiple parts. The data is then transmitted using multiple datagrams having the same group number P, the same tool life L, and the same tool life count type Q.

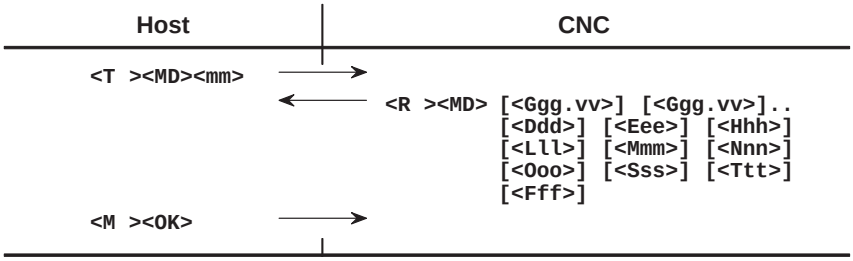


4.2.4.20 Reading modal information for automatic operation

(1) Function

This command reads the previous, current, or next modal data.

(2) Datagram transmission



(3) Parameters

1) mm : Block selection (UI:1, any of 0, 1, and 2)
   0 : Previous block
   1 : Current block
   2 : Next block

2) Ggg.vv : Modal data of G code
   gg : G code group number (UI:2)
   vv : Internal G code (UI:2)
   VV must be specified using two digits.

3) Ddd to Fff : Modal data required in the selected block

```
Ddd | Hhh | Lll  :  UI:4
Mmm              :  UI:3
Nnn              :  UI:5
Ooo | Sss | Ttt  :  UI:4
Eee | Fff        :  UR:9
```

(4) Negative acknowledgment

None

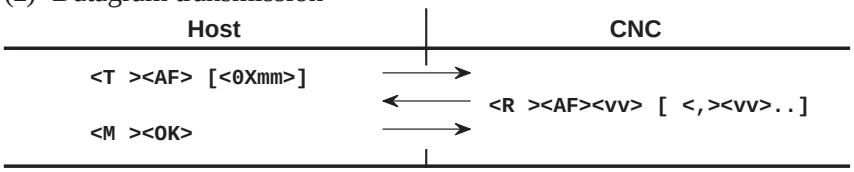
(5) Note

- If no modal data is specified, only the <R ><MD > command is transmitted, and the data section is omitted.
- For internal G code values, see the table in Appendix H.

4.2.4.21 Reading an actual feedrate

(1) Function
This command reads the feedrate of each axis or the composite feedrate of all axes.

(2) Datagram transmission



(3) Parameters

1) mm: Axis request flag (hexadecimal) (HX:4)
This parameter is available for Series 15 only.
For the function of the parameter, see item (3) in Subsection 4.2.4.8.
If mm is omitted, the composite feedrate of all axes is transmitted.

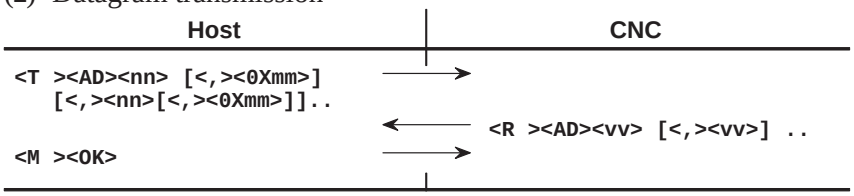
2) vv : Feedrate of each axis (UR:9)
vv is transmitted in ascending order of axis numbers. (1–>2–>3–>...)

(4) Negative acknowledgment
M_NR [0XFFCC] : The axis parameter is invalid.

4.2.4.22 Reading A/D conversion data

(1) Function
This command reads A/D conversion data.

(2) Datagram transmission



(3) Parameters

1) nn : A/D conversion data selection (UI:3)
0 to 1 : General-purpose analog input (Series 15)
0 to 3 : General-purpose analog input (Series 0)
100 : Voltage to which the analog input value of the load current for the 1st spindle is onverted
101 : Voltage to which the analog input value of the load current for the 2nd spindle is converted
200 : Voltage to which the load current for an NC control axis is converted.
The 2nd parameter, 0Xmm, is required for this case.

2) mm: Axis request flag (HX:4)
For the function of this parameter, see item (3) in Subsection 4.2.4.8.

3) vv : Read voltage obtained by conversion (SI:10)
10 * (voltage] 128)/128 volts

(4) Negative acknowledgment

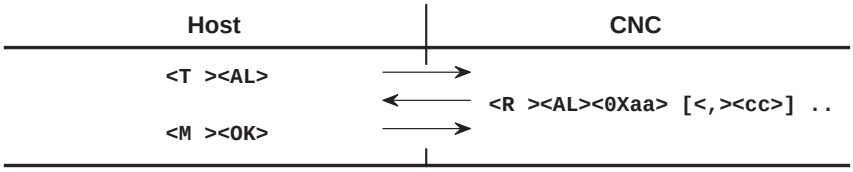
1) **M_NR [0XFFC9]** : The axis request flag is omitted.
2) **M_NP [0XFB94]** : Alarm status (for Series 0/16/18/21)

4.2.4.23 Reading alarm information

(1) Function

This command reads alarm information.

(2) Datagram transmission



(3) Parameters

aa : Alarm information (HX:4)

Bit

0 : Background PS alarm
(PS alarm stands for program setting alarm.)
1 : Foreground PS alarm
2 : Overheat alarm
3 : Not determined
4 : Not determined
5 : SW alarm (parameter input enable)
6 : OT alarm (overheat label/spindle alarm)
7 : PMC error
8 : External alarm
9 : Not determined
10 : Fatal PS alarm
11 : Not determined
12 : Servo alarm
13 : I/O alarm
14 : PW alarm (Power off request)
15 : Not determined

(4) Negative acknowledgment

None

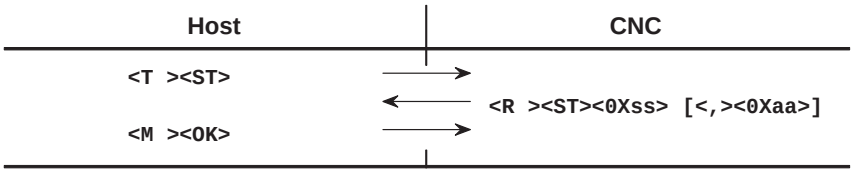
4.2.4.24 Reading status information

(1) Function

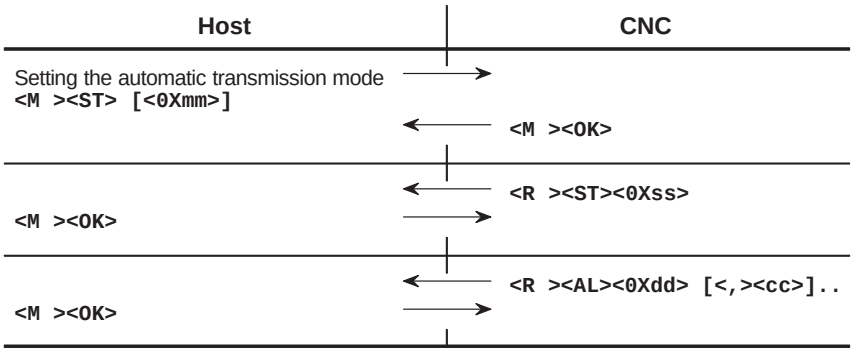
This command reads status information.

(2) Datagram transmission

1) Request from the host computer



2) Automatic transmission to the host



(3) Parameters

1) mm: Mask bit for information to be transmitted (HX:4)
Each bit corresponds to the ss bit.
1 : Masked (not transmitted)
0 : Not masked (transmitted)
The initial value is 0XFFFF.
If mm is omitted, all the bits are enabled (transmitted).

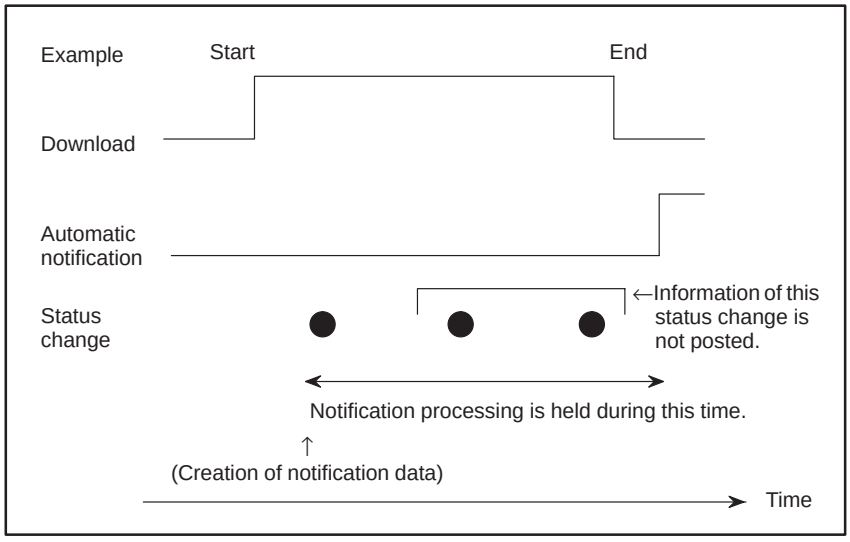
2) ss : CNC status (HX:4)
Bit
0 : Rewind (RWD)
1 : Alarm (AL)
2 : Rest (RST)
3 : Automatic operation currently stopped (SPL)
4 : Automatic operation currently started (STL)
5 : Automatic operation in process (OP)
6 : Servo ready (SA)
7 : CNC ready (MA)
8 : Not used
9 : Not used
10 : Not used
11 : Not used
12 : M00 output (M00)
13 : M01 output (M01)
14 : M02 output (M02)
15 : M30 output (M30)

3) aa : Alarm information (HX:4)
For the alarm information, see the parameter described in Subsection 4.2.4.23.
When AL of code ss is 1, alarm status code aa is transmitted after the code ss.

4) dd : Alarm information (HX:4)
Value
0 : Background PS alarm
1 : Foreground PS alarm
2 : Overheat alarm
3 : Not determined
4 : Not determined
5 : SW alarm (parameter input enable)
6 : OT alarm (overheat label/spindle alarm)
7 : PMC error
8 : External alarm
9 : Not determined
A : Fatal PS alarm
B : Not determined
C : Servo alarm
D : I/O alarm
E : PW alarm (Power off request)
F : Not determined
10 : Battery alarm

(4) Negative acknowledgment
None

(5) Notes
1) In automatic notification mode, <R ><ST> or <R ><AL> is automatically transmitted from the CNC.
<R ><AL> is transmitted only when an alarm is issued.
2) In automatic notification mode, no commands other than <M><ST> can be transmitted from the host.
3) When the status is updated while the host is issuing another request (such as a request to download a program), information about the first status change is posted to the host by automatic notification from the CNC after the termination of the preceding request.
This is because the DNC2 protocol cannot handle more than one request at any one time. So, automatic notification processing is held during processing of the preceding request.



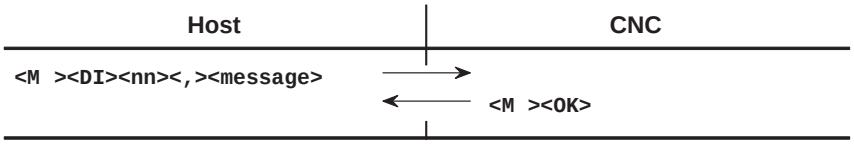
Reading status command can be transmitted from the host when accurate notification of status change is needed.

4.2.4.25 Operator messages

(1) Function

This command displays the operator messages on the Screen of the CNC.

(2) Datagram transmission



(3) Parameters

1) nn : Message number (SI:1 / –1 to –5 or 1 to 5)

Five different messages can be displayed at a time because the CNC has five message areas.
If nn is negative, the old message on the Screen is cleared, and a new message stored in the first area is displayed first. If nn is positive, a new message is added and displayed after the old message on the Screen.
Each message area can contain up to 128 characters.

2) message : Message in ASCII code to be displayed on the Screen (CH:32)

Up to 32 characters of the message can be displayed at a time.

(4) Negative acknowledgment

M_NR [0XFFC9] : The message number is incorrect.

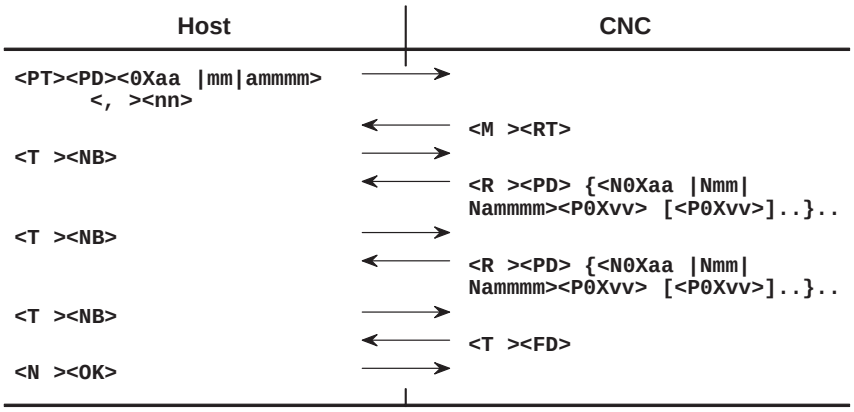
4.2.4.26 Reading PMC data

(1) Function

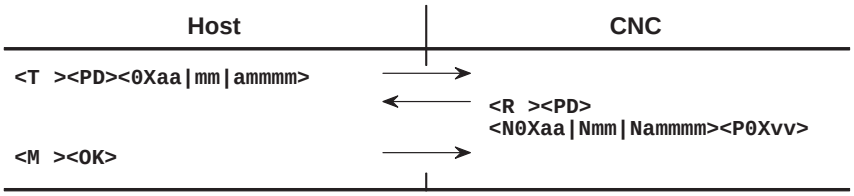
Data in the PMC area is read.

(2) Datagram transmission

a) Type A



b) Type B



(3) Parameters

1) aa : PMC address (HX: 8) (applicable to the Series 15 PMC–NA only (address specification type))
The address is specified with eight hexadecimal characters by prefixing 0X41 to the 6–digit physical address, given in the list of relationships between PMC addresses and physical addresses in Section 3.3.2 of the “FANUC PMC–MODEL N Programming Manual” (B–61013E–2).
Example: For F000, specify 0X41FFE200.

mm : PMC data number (UI:4) (applicable to the Series 0 only (data number specification type))
Refer to “FANUC Series 0–MC Connection Manual”(B–61393E) and “FANUC PMC MODEL – K/L/M Programming Manual”(B–55193E).
Specify the numeric part of the PMC address, which is the PMC address less the initial alphabetic character.
Example: For F148, specify 148.
For R700, specify 700.

ammmm : PMC data number (<a> <mmmm>) (Series 16/18, and PMC–NA, NB, NB2 of Series 15 (data number specification type))

a : PMC data address (UI: 1)
0 → A 3 → F 6 → R 9 → Y
1 → C 4 → G 7 → T
2 → D 5 → K 8 → X

mmmm : PMC data number (UI: 4)
Example: For D500, specify 20500.

NOTE
The PMC data number must consist of five numeric characters if address <a> is a non–zero value.

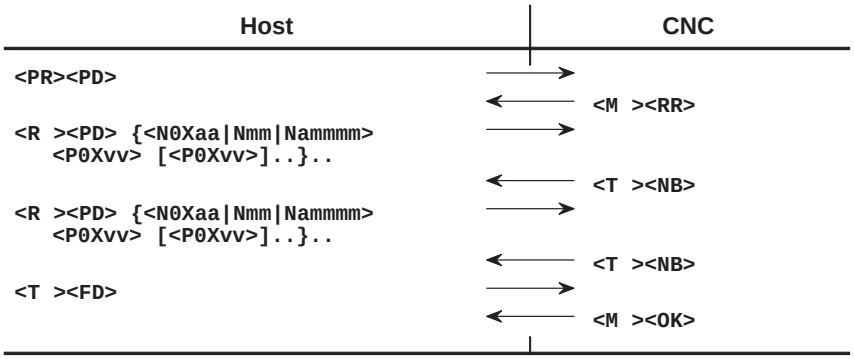
2) nn : Number of bytes to be read from address aa (UI:2/1 nn 32)
3) NOXaa : Starting address of read data (POXvv) (HX: 8)
4) POXvv : PMC data value (HX: 2)
The value is posted for each byte.

(4) Negative acknowledgment
M_NR [0XFFCA]:
The PMC address or PMC data number is invalid.

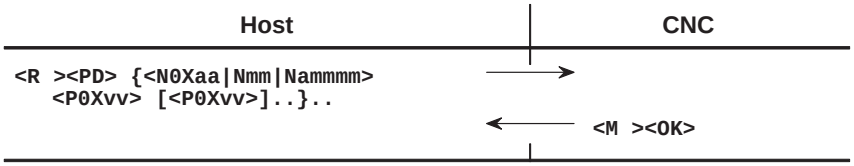
4.2.4.27 Writing PMC data

(1) Function
Data is written into the PMC area.
The host computer communicates with the PMC application software.
FANUC does not define the meanings of the addresses in RAM. These depend on the settings made between the host computer and CNC.

(2) Datagram transmission
a) Type A



b) Type B



(3) Parameters
1) aa : PMC address (HX: 8) (applicable to the Series 15 PMC–NA only (address specification type))
The address is specified with eight hexadecimal characters by prefixing 0X41 to the 6–digit physical address, given in the list of relationships between PMC addresses and physical addresses in Section 3.3.2 of the “FANUC PMC–MODEL N Programming Manual” (B–61013E–2).
Example: For F000, specify 0X41FFE200.

mm : PMC data number (UI: 4) (applicable to the Series 0 only (data number specification type))

Refer to “FANUC Series 0–MC Connection Manual” (B–61393E) and “FANUC PMC MODEL–K/L/M Programming Manual” (B–55193E).
Specify the numeric part of the PMC address, which is the PMC address less the initial alphabetic character.
Example: For F148, specify 148.
For R700, specify 700.

ammmm : PMC data number (<a> <mmmm>) (Series 16/18, and PMC–NA, NB, NB2 of Series 15 (data number specification type))

a : PMC data address (UI: 1)

0 → A	3 → F	6 → R	9 → Y
1 → C	4 → G	7 → T	
2 → D	5 → K	8 → X	

mmmm : PMC data number (UI: 4)
Example: For D500, specify 20500.

NOTE
The PMC data number must consist of five numeric characters if address <a> is a non–zero value.

2) POXvv: Byte written in the PMC RAM (HX: 2)

(4) Negative acknowledgment

1) **M_NR [0XFFCA]**:
The PMC address or PMC data number is invalid.

2) **M_NR [0XFBA3]**:
The PMC address or PMC data number is beyond the allowable range.

(5) Note

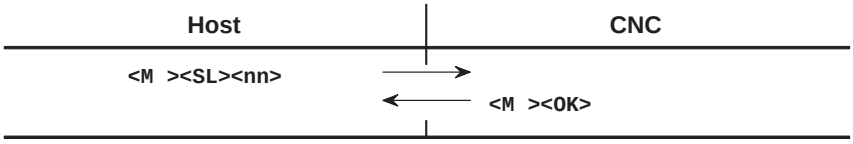
It is recommended that the data access direction for each PMC RAM be determined explicitly between the host computer and PMC. Otherwise, when the host computer and PMC attempt to write data to the same RAM address at the same time, a contention occurs (if two units attempt to write the same address at the same time, the write processing being performed by one of the two units will fail).

4.2.4.28 Selecting a part program

(1) Function

This command selects a part program on tape storage.

(2) Datagram transmission



(3) Parameter

nn : Part program number (UI:4)

(4) Negative acknowledgment

1) **M_NR [0XFC0C]** :
Program nn is not found.

2) **M_NR [0XFC08]** :
Alarm status (for Series 0 only), or the program is running.

3) **M_NR [0XFC09]** :
The system is not in the AUTO mode. (For Series 0 only)

(5) Note

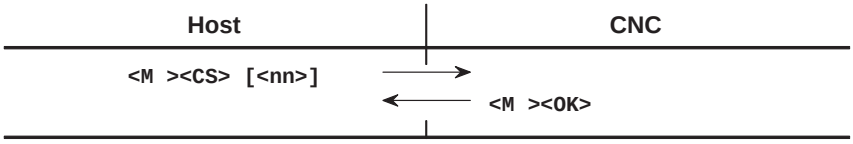
With the Series 15, only memory mode (MEM) and edit mode (EDIT) are valid. In other modes, the sequence terminates normally, but program selection is not performed.

4.2.4.29 Executing a program

(1) Function

This command selects and executes a program.

(2) Datagram transmission



(3) Parameter

nn : Part program number (UI:4)

If this parameter is specified, the CNC selects part program nn and executes it. If it is omitted, the CNC executes the currently selected part program.

(4) Negative acknowledgment

1) **M_NR [0XFC0C]** :
Program nn is not found.

2) **M_NR [0XFC09]** :
The mode is incorrect. (Neither AUTO mode nor MEM mode)

3) **M_NR [0XFC0A]** :
Program nn is already being executed, or the CNC is in the alarm status.

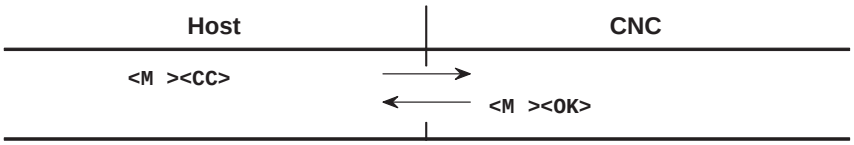
4) **M_NR [0XFC08]** :
Selecting the program was rejected.
The command for selecting program nn is issued again during automatic operation.

4.2.4.30 Reset

(1) Function

This command resets the CNC. It functions in the same way as the external reset (ERS) signal of the machine interface.

(2) Datagram transmission



(3) Parameters

None

(4) Negative acknowledgment

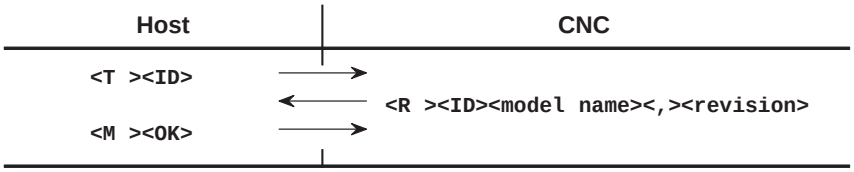
None

4.2.4.31 Reading a system ID

(1) Function

This command reads information on the system ID.

(2) Datagram transmission



(3) Parameters

1) model name : CNC model name
   "F15M9A/F15M14/F15T9A/F15T14" (Series 15)
   "FS0–MC/FS0–TC" (Series 0)
   "F16–MA/F16–TA/F18–MA/F18–TA" (Series 16/18–A)
   "F16–MB/F16–TB/F18–MB/F18–TB" (Series 16/18–B)
   "F16–MC/F16–TC/F18–MC/F18–TC" (Series 16/18–C)
   "F16*i*–MA/F16*i*–TA/F18*i*–MA/F18*i*–TA/ F21*i*–MA/F21*i*–TA" (Series 16*i*/18*i*/21*i*–A)
2) revision : CNC DNC2 software revision
   "1.0" (Series 15)
   "1.1" (Series 0/16/18/21)

(4) Negative acknowledgment

None

4.3 DATAGRAM PROTOCOL

This section describes the datagram protocol between the FANUC CNC and the host computer.

4.3.1 Error Handling

The DNC2 has six negative acknowledgments and one interrupt command for handling errors.

4.3.1.1 Negative acknowledgment

(1) Syntax

<T_BD | M_ER | M_NR | M_NP | T_NP | M_IL> [<0Xeeee>]

(2) Functions

T_BD: Sent if the datagram protocol is incorrect. See Appendix D.
M_ER: Sent if a syntax error is detected in the datagram.
M_NR: Sent if the number given in the data section of the preceding datagram is incorrect.
M_NP: Sent if executing commands in the preceding datagram is rejected.
T_NP: Sent if the CNC cannot access the requested data.
M_IL: Sent if errors other than the above occur.

(3) Error code

eeee : This four-digit hexadecimal code indicates the cause of the error. The parameter specifies whether the code is output. See Appendixes B and F.

(4) Protocol

The receiving device sends a negative acknowledgment to the sending device. The negative acknowledgment indicates that an error occurred in the receiving terminal.
The device that sent the negative acknowledgment and the device that received it must interrupt transmission immediately and return to the initial state.

4.3.1.2 Interrupt command

(1) Syntax

<T_BD>

(2) Function

The interrupt command forcibly terminates processing currently being transmitted. Either the CNC unit or the host computer can send this command at any time.

(3) Protocol

As soon as the terminal receives the interrupt command, it interrupts transmission and returns to the initial state. When the terminal is in the initial state, it ignores the interrupt command. This command is used to forcibly return the application layer to the initial state.

4.3.2 Interrupt Procedure

4.3.2.1 Interrupt

The datagram transmission procedure can generally be started in the idle state.
Unless both the CNC and the host terminate the current datagram transmission procedure, they cannot start a new transmission procedure. In other words, they cannot interrupt the datagram transmission procedure as shown in Fig. 4.3.2.1.

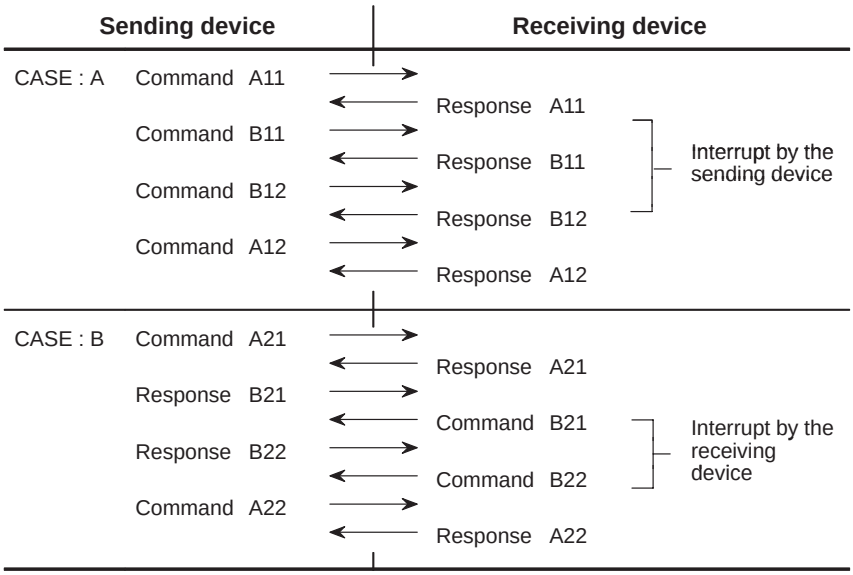


Fig.4.3.2.1 Example of Interrupts

III. COMMUNICATION PROGRAM LIBRARY

1 OUTLINE

This manual explains the use and installation of the communication program library for a personal computer. They are specified in the DNC2 protocol developed for DNC communication between a FANUC CNC and personal computer (PC), or between a CNC and host computer.

This library cannot be used on its own. Be sure to create application programs and link them with this library. For creating and linking application programs, see Section 3.3.

The demonstration program (sample application program) supplied with this system permits simple control of the CNC.
For use of the demonstration program, see Appendix K.

2 CONFIGURATION

IBM PC series

Model : PC, AT, PS/2
OS : PC–DOS Ver. 3.1 or later
Compiler: MS–C compiler Ver. 5.0 or later
Memory : The communication package library alone uses 134K bytes of memory.
Others : Use of the demonstration program requires the CGA interface.

NEC PC–98 series

Model : Model VM and all subsequent models (except the high–resolution mode)
OS : MS–DOS Ver. 3.1 or later
Compiler: MS–C compiler Ver. 5.0 or later
Memory : The MS–DOS 3.1 system alone uses 60K bytes of memory. The MS–DOS 3.3B system alone uses 89K bytes of memory. The communication package library alone uses 134K bytes of memory.
Others : Only the RS–232–C standard port is supported.

Fujitsu FMR series

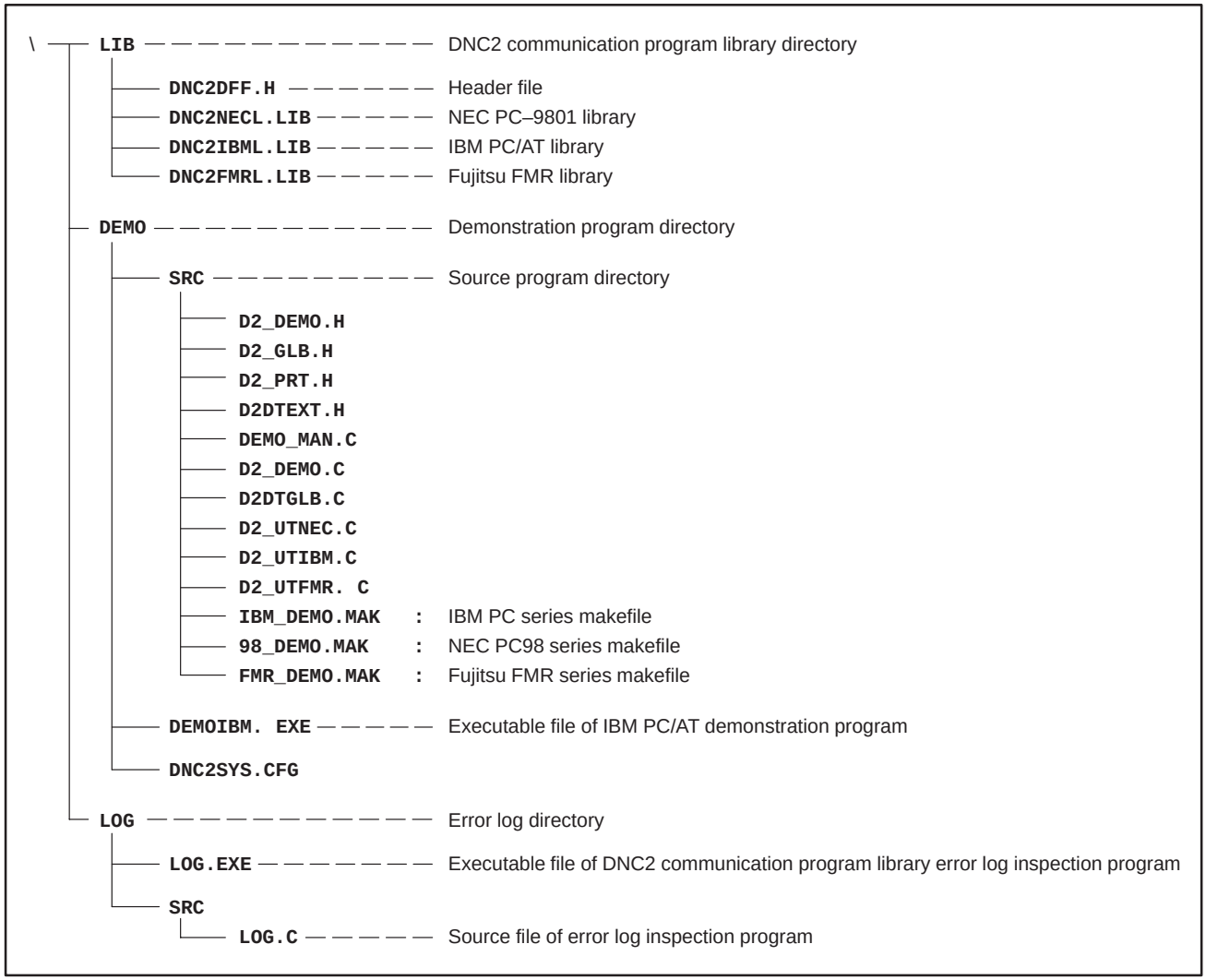
Model : FMR–50/60/70
OS : MS–DOS Ver. 3.1 or later
Compiler: MS–C compiler Ver. 5.0 or later
Memory : The communication package library alone uses 134K bytes of memory.
Others : The demonstration program is designed for high resolution

NOTE
Memory requirements vary somewhat depending on whether the config.sys file contains the specifications of files and buffers. Memory requirements are also increased by the size of device drivers such as print.sys, if included.

3 INSTALLATION

3.1 SUPPLIED PROGRAMS

The FANUC DNC2 communication program library is stored in one volume of floppy disk shown below. The contents of the floppy disk are shown in the figure below.
The libraries are all created for large models.



The programs stored in the DEMO directory are designed with the Series 0–MC specified as the CNC, to make the user familiar with DNC2. These programs do not run normally even when compiled and linked.
Use these programs merely for reference to understand the use of the DNC2 communication library.

3.2 INSTALLATION PROCEDURE

This section explains the procedure for installing the library.

Copy the header file and the library for the model to be used to the directory (the directory to be compiled) in which application files exist.

The following shows the method for copying files from the floppy disk in drive A to the DNC2 directory in the hard disk in drive C.

```
C>CD \DNC2
C>COPY A: \LIB \DNC2DEF. H C:
C>COPY A: \LIB \DNC2IBML. LIB C:
```

NOTE
For IBM PC/AT, replace the yen signs with backslashes.

3.3 COMPILATION AND LINKING

(1) Compiler

The object compiler is version 5.0 or a later version of the MS–C compiler produced by Microsoft.

(2) Application program setup

This library only executes single functions (commands). It requires a main program which executes different types of functions in combination. The main program is called an application program.

When creating an application program, be sure to include the header file DNC2DEF. H at the beginning. (Insert the following line at the beginning of each file.)

```
#include "dnc2def. h"
```

If the header file is not included, functions in this library cannot be called correctly.

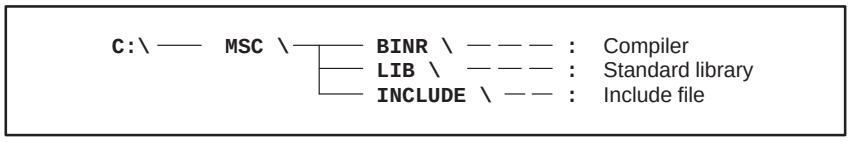
(3) Setting environment variables

The following three environment variables must be set for compilation to start normally. (For environment variables, refer to the PC–DOS or MS–DOS manual.)

- PATH : Directory in which the compiler exists
- LIB : Directory in which the standard library of the compiler exists
- INCLUDE : Directory in which the include file of the compiler exists

Check that these environment variables are set correctly.

Set environment variables as follows if the directory structure is as shown below.




```
C>SET PATH = C: \MSC \BINR
C>SET LIB  = C: \MSC \LIB
C>SET INCLUDE = C: \MSC \INCLUDE
```

(4) Compilation and linking

For large models, an application program is compiled as follows, where the name of the file containing the application program is <APLI. C>.

```
C>CL –AL APLI. C DNC2IBML. LIB
```

After compilation, the application program is automatically linked and an executable file <APLI. EXE> is generated.

For details of compiler options and compilation, refer to the MS–C compiler manual.

4 FUNCTION SPECIFICATIONS

4.1 USE

Call functions in the calling format shown in Section 4.4. If the types of arguments agree, it does not matter whether the variable names differ.

The return values of functions are always of the short type. As required, convert them to a proper type (cast) for use.

Example)

```
test( )
{
short ans;
char ch;
short ax;
bouble dt [15];
          :
          :
      ch = 0;
      bit = 0x01;
          :
          :
      if (( ans = D_servo (ch, bit, &dt [0])) << 0) {
               /* Error handling */
               if (ans == –1) {
                   :
               } else if (ans == –2) {
                   :
                   :
               }
      }
       :
       :
}
```

4.2 NOTES

(1) These functions do not return until communication with the NC unit terminates completely.
Note that downloading/uploading or other time–consuming processes disable other processes.

(2) Specify in uppercase a program number or any other character string used as an argument to call a function.
Message output can be input in both uppercase and lowercase.

(3) Specify the name of an input–output file with an absolute path or a path relative to a current directory.

(4) Since the functions that output results to files open files in the overwrite mode, the contents of existing files are deleted.

(5) Before calling functions that output results to memory, allocate a free space large enough to store results. The functions assume that an area for storing results is allocated.

(6) Arguments selected by numbers (mode of D_tool_pos()) can be specified directly by numeric values, or by defined names. (The latter is recommended.)
When specifying arguments by defined names, be sure to include dnc2def.h.

Example)
```
#include ”dnc2def.h”
...
D_tool_pos(0,MACHINE_POS,0x13,&tooldat [0 ] );
```

(7) The types of variables used in the function specifications are described below.

char	: One–byte signed integer or 7–bit character
uchar	: One–byte unsigned integer or 8–bit character
short	: Two–byte signed integer
ushort	: Two–byte unsigned integer
long	: Four–byte signed integer
double	: Eight–byte signed real number

4.3 FUNCTIONS

Functions of the communication package are shown below.

No.	Function	Description	Send command
1	**D_download**	Part program download	PRPM
2	**D_download_wait**	Waits for download request and starts downloading a part program.	PTPM wait
3	**D_upload**	Part program upload	PTPM
4	**D_delete**	Deletes part programs.	MCPM
5	**D_fdir**	Reads part program directory information for one program.	LIPM
6	**D_alldir**	Reads part program directory information for all programs (file type).	LIPM
7	**D_alldir_mem**	Reads part program directory information for all programs (memory type).	LIPM
8	**D_freemem**	Reads the size of the free area	T FR
9	**D_tool_pos**	Reads the tool position.	T MP T WP T SP
10	**D_servo**	Reads the servo delay.	T SE
11	**D_AccDec**	Reads the acceleration/deceleration delay.	T AE
12	**D_diagnose**	Reads the machine interface signal.	T MI
13	**D_exec_num**	Reads the numbers of the program and block being executed.	T PN T SN
14	**D_parameter_r**	Reads CNC parameters.	PTPA
15	**D_parameter_w**	Writes CNC parameters (file type).	PRPA
16	**D_parameter_w_mem**	Writes CNC parameters (memory type).	PRPA
17	**D_pitch_r**	Reads pitch error data (file type).	PTPE
18	**D_pitch_r_mem**	Reads pitch error data (memory type).	PTPE
19	**D_pitch_w**	Writes pitch error data (file type).	PRPE
20	**D_pitch_w_mem**	Writes pitch error data (memory type).	PRPE
21	**D_offset_r**	Reads the tool offset (file type).	PTTO
22	**D_offset_r_mem**	Reads the tool offset (memory type).	PTTO
23	**D_offset_w**	Writes the tool offset (file type).	PRTO
24	**D_offset_w_mem**	Writes the tool offset (memory type).	PRTO
25	**D_variable_r**	Reads custom macro variables (file type).	PTMV
26	**D_variable_r_mem**	Reads custom macro variables (memory type).	PTMV
27	**D_variable_w**	Writes custom macro variables (file type).	PRMV
28	**D_variable_w_mem**	Writes custom macro variables (memory type).	PRMV
29	**D_toollife**	Reads the tool life management data (file type).	PTTL
30	**D_toollife_mem**	Reads the tool life management data (memory type).	PTTL
31	**D_modal**	Reads automatic operation continuous–state information (file type).	T MD
32	**D_modal_mem**	Reads automatic operation continuous–state information (memory type).	T MD
33	**D_actualspeed**	Reads the actual speed.	T AF
34	**D_ADconcert**	Reads A/D conversion data.	T AD
35	**D_alarm**	Reads alarm information.	T AL
36	**D_status**	Reads status information.	T ST
37	**D_message**	Message output	T DI
38	**D_PMCdata_r**	Reads PMC data.	PTPD T PD
39	**D_PMCdata_w**	Writes PMC data.	PRPD R PD
40	**D_select**	Searches for a program number.	M SL
41	**D_start**	Cycle start	M CS
42	**D_reset**	Reset	M CC
43	**D_systemID**	Reads system ID.	T ID
44	**D_initial**	Initializes the communication port.	–

4.4 DETAILS

This section explains the functions in detail.

For an explanation of the return value (negative acknowledgment) specific to each function, see Appendix J.

The functions correspond to the service functions explained in Section 4.2.4 of II–4 almost on a one–to–one basis. Therefore, see Section 4.2.4.

1) Function : **D_download** Part program download
   Calling format : **short D_download (char ch_no, uchar *file_name, char *prg_no)**
   Arguments : **char ch_no**
   Cannel number (0 to 3) <numeric value>
   uchar *file_name
   Name of file being downloaded
   <haracter string>
   char *prg_no
   Program number (o0001 to o9999)
   <character string>
   Return value : **= 0** Normal
   = FFFF Communication error
   = FFFE File open error
   = FFFD File read error
   = FFF0 Sequence error
   Others See Appendix F.
   Outline : Transfers part programs from a host computer to the CNC.

   * Release background edit mode and simultaneous edit mode (Series 15 only) in advance.

2) Function : **D_download_wait**
   Waits for download request and starts download of part program.
   Calling format : **short D_download_wait (char ch_no)**
   Arguments : **char ch_no**
   Channel number (0 to 3) <numeric value>
   Return value : **= 0** Normal
   = FFFF Communication error
   = FFFE File open error
   = FFFD File read error
   = FFF0 Sequence error
   Others See Appendix F.
   Outline : Waits for a download request from the CNC.
   On receiving a download request, determines the file name as <0xxxx.PRG> from the requesting program number and starts download. (For example, if the program number is 1, the file name is <O0001.PRG>.)
   This function terminates after downloading one file.
   This function is used to start downloading when a CNC cycle is started as in DNC operation.

   The general procedures for DNC operation are explained below.

   (1) Put the system in the tape mode (for Series 15 only)

   (2) Use the PMC data read function to check that the CNC has become ready.

(3) Select the program to perform DNC operation using D_select. (Series 15)
For Series 0, set the number of the program to perform DNC operation by the MAP parameter screen DNC FILE SELECTION in advance.

(4) Issue the instructions for starting the cycle to the CNC by the PMC data write function. The PMC ladder program monitors the directions and starts the CNC cycle. (Although the cycle can also be started by D_start, this method is better as it is possible to check whether the machine has been set up.)

(5) Call this function. Control is returned from the function when DNC operation terminates normally or abnormally.

Perform 5 as soon as possible after 4. Otherwise, a timeout will occur.

3) Function : **D_upload** Part program upload

Calling format : **short D_upload (char ch_no, uchar *file_name, char *prg_no)**

Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
uchar *file_name
Name of file being uploaded
<character string>
char *prg_no
Program number (o0001 to o9999)
<character string>

Return value :
= 0 Normal
= FFFF Communication error
= FFFE File open error
= FFFC File write error
= FFF0 Sequence error
Others See Appendix F.

Outline : Transfers part programs from a host computer to the CNC.

Release background edit mode and simultaneous edit mode (Series 15 only) in advance.

4) Function : **D_delete** Deletes part programs.

Calling format : **short D_delete (char ch_no, char *prg_no)**

Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
char *prg_no
Program number (o0001 to o9999)
<character string>
[o**** for all programs]

Return value :
= 0 Normal
= 1 No program (for Series 15 only)
= FFFF Communication error
= FFF0 Sequence error
Others See Appendix F.

Outline : Directs the CNC to delete part programs.
(The specified programs or all programs are deleted.)

Note : If all programs are deleted when the CNC is FANUC Series 15, the selected programs are not deleted unless the system is in the edit mode. A return value indicating abnormal termination is returned.
The recommended method for deleting all part programs is to change to the edit mode using the PMC data write function, delete the part programs, then return to the original mode, or to select a small dummy program for deletion.

Release background edit mode and simultaneous edit mode (Series 15 only) in advance.

5) Function : **D_fdir** Reads part program directory information for one program.

Calling format :
```
short D_fdir
(char ch_no, char *prg_no)
```

Arguments :
char ch_no
Channel number (0 to 3) <numeric value>
char *prg_no
Program number (o0001 to o9999)
<character string>

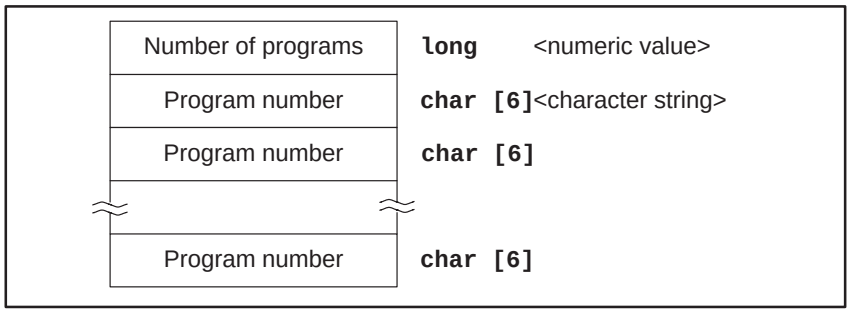
Return value :
= 0 The specified program exists.
= FFFF Communication error
= FFF0 Sequence error
Others See Appendix F.

Outline : Reads a part program from the CNC and checks that the program exists.

6) Function : **D_alldir** Reads part program directory information for all programs.

Calling format :
```
short D_alldir
(char ch_no, uchar *file_name)
```

Arguments :
char ch_no
Channel number (0 to 3) <numeric value>
uchar *file_name
Directory storage file name



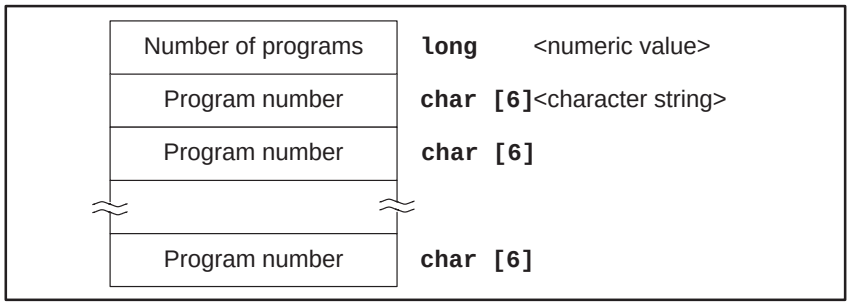
Return value :
= 0 The specified program exists.
= FFFF Communication error
= FFFE File open error
= FFFC File write error
= FFF0 Sequence error
Others See Appendix F.

Outline : Reads all part programs from the CNC and writes the program numbers to a file.

7) Function : **D_alldir_mem** Reads part program directory information for all programs.

Calling format : **short D_alldir_mem (char ch_no, long size, char *data)**

Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
long size Buffer size <numeric value>
char *data Directory storage buffer



Return value : **= 0** Program exists.
= FFFF Communication error
= FFF0 Sequence error
Others See Appendix F.

Outline : Reads all part programs from the CNC and stores the program numbers in the buffer.

8) Function : **D_freemem** Reads the size of the free area.

Calling format : **short D_freemem (char ch_no, long *size)**

Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
long *size
Area for storing free area size <numeric value>

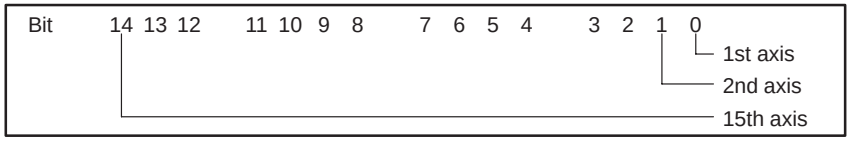
Return value : **= 0** Normal
= FFFF Communication error
= FFF0 Sequence error
Others See Appendix F.

Outline : Directs the CNC to read the size of the free area and returns the size.

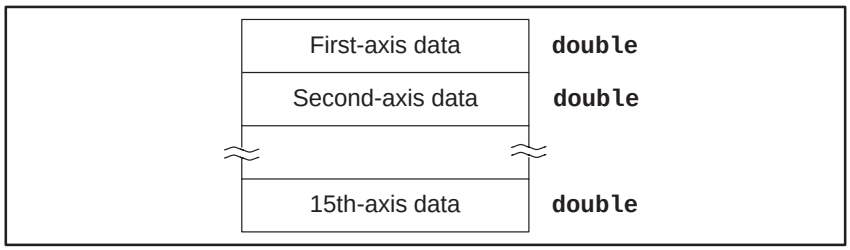
9) Function : **D_tool_pos** Reads the tool position.

Calling format : **short D_tool_pos (char ch_no, char mode, short axis, double *data)**

Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
char mode
1 [WORK_POS] <numeric value>
Reading of absolute position
2 [MACHINE_POS] Reading of machine position
3 [SKIP_POS] Reading of skip position
short axis
Axis specification <numeric value>
(Turns on the bits corresponding to axes. Bits 0 to 14.)



double *data
Tool position data storage buffer



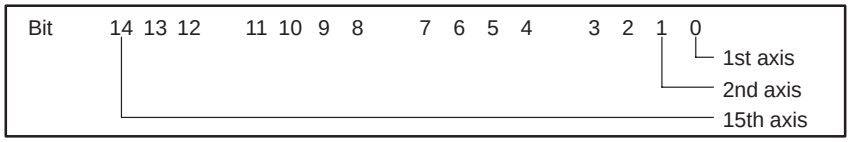
Return value : **= 0** Normal
= FFFF Communication error
= FFF0 Sequence error
Others See Appendix F.

Outline : Reads either the absolute position, machine position, or skip position, and stores the data for the specified axes in the corresponding positions. Data of axes that are not specified is undefined.

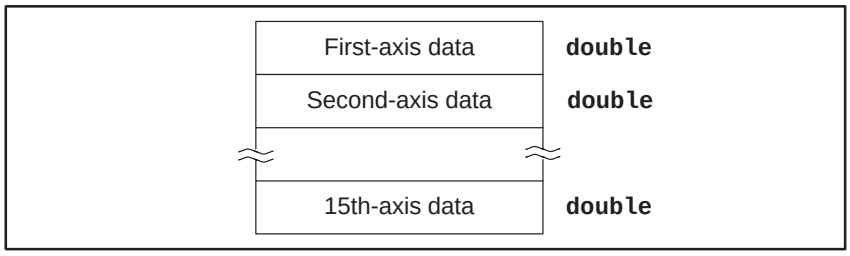
10) Function : **D_servo** Reads the servo delay.

Calling format : **short D_servo (char ch_no, short axis, double *data)**

Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
short axis
Axis specification <numeric value>
(Turns on the bits corresponding to axes. Bits 0 to 14.)



double *data Servo delay buffer



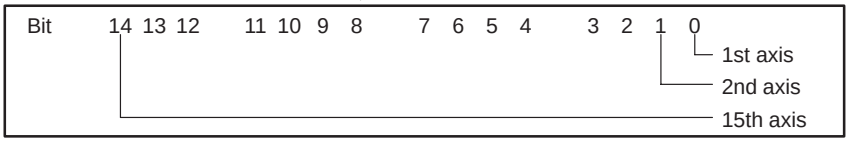
Return value : **= 0** Normal
= FFFF Communication error
= FFF0 Sequence error
Others See Appendix F.

Outline : Reads the servo delay and stores the data of the specified axes in the corresponding positions. Data of axes that are not specified is undefined.

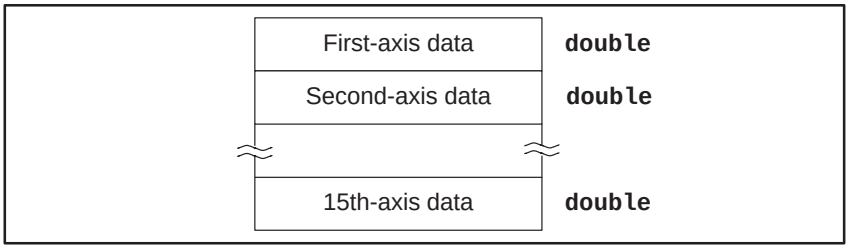
11) Function : **D_AccDec** Reads the acceleration/deceleration delay.

Calling format : **short D_AccDec (char ch_no, short axis, double *data)**

Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
short axis
Axis specification <numeric value>
(Turns on the bits corresponding to axes. Bits 0 to 14.)



double *data Servo delay buffer



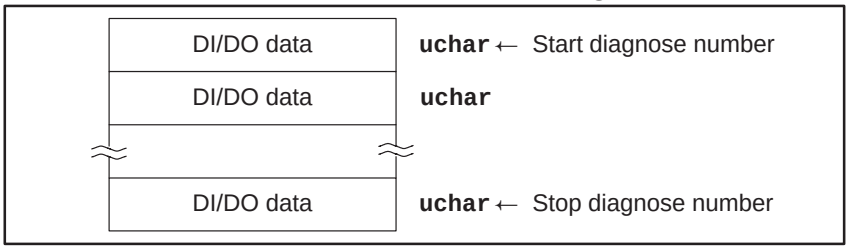
Return value : **= 0** Normal
= FFFF Communication error
= FFF0 Sequence error
Others See Appendix F.

Outline : Reads the acceleration/deceleration delay and stores the data of the specified axes in the corresponding positions. Data of axes that are not specified is undefined.

12) Function : **D_diagnose** Reads machine interface signal.

Calling format : **short D_diagnose (char ch_no, short start_no, short stop_no, uchar *data)**

Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
short start_no
Start diagnose number (1 to 999)
<numeric value>
short stop_no
Stop diagnose number <numeric value>
(0 to 999. When 0, omit this argument.)
uchar *data DI/DO signal data buffer



Return value : **= 0** Normal
= FFFF Communication error
= FFFB Diagnose number error (M_NR)
= FFF0 Sequence error
Others See Appendix F.

Outline : Reads the machine interface signal data and returns data from the start of operation until it stops. If a stop diagnose number is omitted, only the data corresponding to the start diagnose number if returned.

13) Function : **D_exec_num** Reads the number of the program or block being executed.

Calling format : **long D_exec_num (char ch_no, char type, long *number)**

Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
char type
1 [PROG_NUM] Program number
<numeric value>
2 [SEQ_NUM] Sequence number
long *number
Program/sequence number storage buffer
<numeric value>

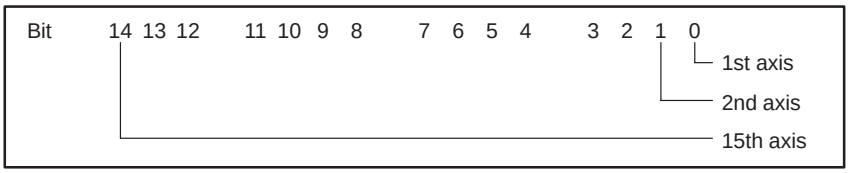
Return value : **= 0** Normal
= FFFF Communication error
= FFF0 Sequence error
Others See Appendix F.

Outline : Reads the number of the program or block being executed by the CNC and returns it. If no program number or sequence number is defined or no program or block is being executed, zero is returned.

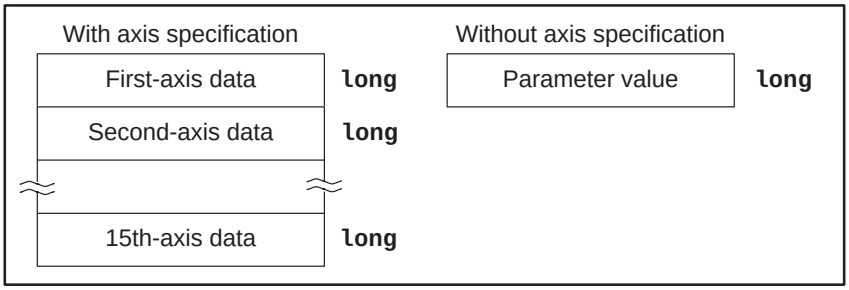
14) Function : **D_parameter_r** Reads CNC parameters.

Calling format : **short D_parameter_r (char ch_no, short p_no, short axis, long *data)**

Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
short p_no
Parameter number <numeric value>
short axis
=0 Without axis specification
<numeric value>
≠0 With axis specification
(Turns on the bits corresponding to axies. Bits 0 to 14.)



long *data Servo delay buffer



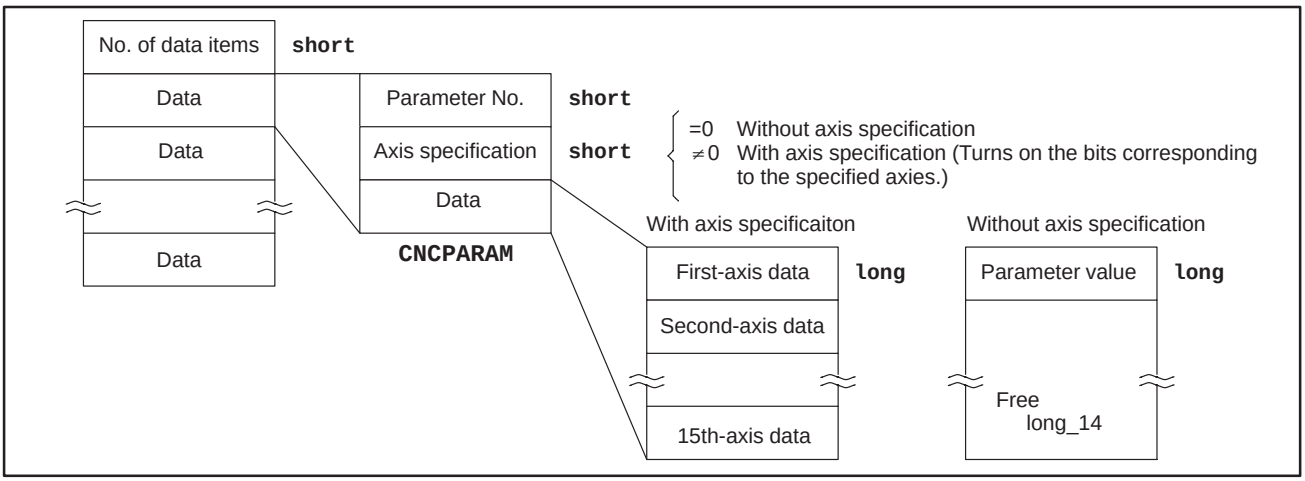
Return value : **= 0** Normal
= FFFF Communication error
= FFF0 Sequence error
Others See Appendix F.

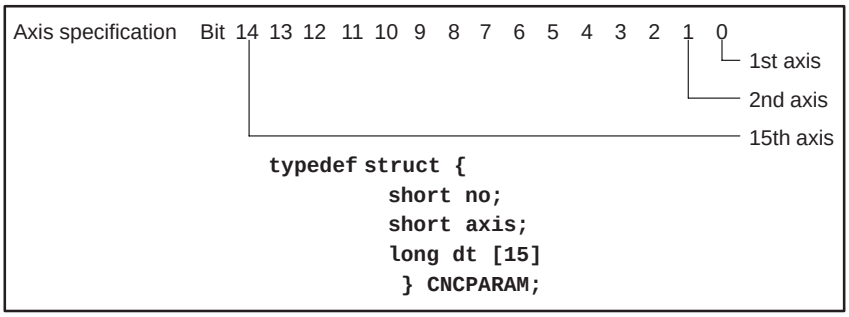
Outline : Directs the CNC to read CNC parameters and returns the parameter or the parameters corresponding to the specified axes. Data of axes that are not specified is undefined.

15) Function : **D_parameter_w** Writes CNC parameters.

Calling format :
```
short D_parameter_w
(char ch_no, uchar *file_name)
```

Arguments :
char ch_no
Channel number (0 to 3) <numeric value>
uchar *file_name
Name of write parameter file



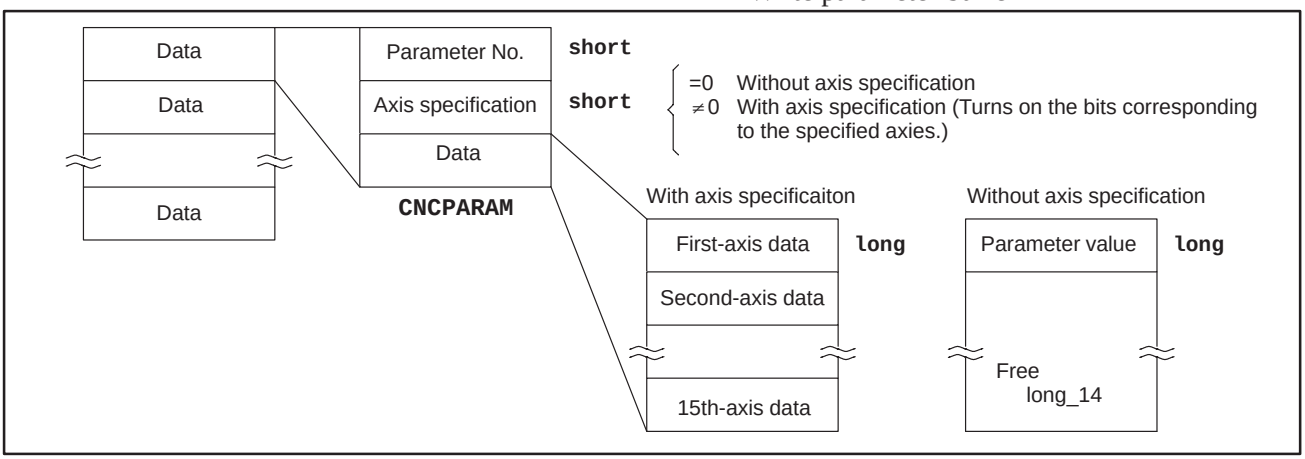


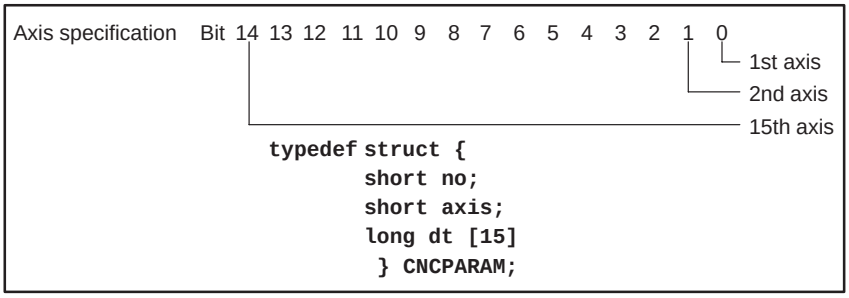
Return value : **= 0** Normal
= FFFF Communication error
= FFFE File open error
= FFFD File read error
= FFF0 Sequence error
Others See Appendix F.

Outline : Writes the parameters stored in the file or the parameters corresponding to the specified axes to the CNC. Data of axes not specified is ignored.

Note : For the effective timing of updating parameters, refer to the CNC operator's manual.

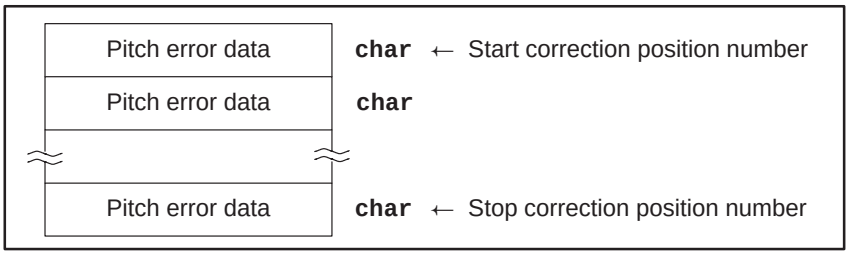
16) Function : **D_parameter_w_mem** Writes CNC parameters.
Calling format : **short D_parameter_w_mem (char ch_no, short cout, CNCPARAM *data)**
Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
short count
Parameter data count <numeric value>
CNCPARAM *data
Write parameter buffer





Return value : **= 0** Normal
= FFFF Communication error
= FFF0 Sequence error
Others See Appendix F.
Outline : Writes the parameter stored in the buffer or the parameters corresponding to axes to the CNC. Data of axes not specified is ignored.
Note : For the effective timing of updating parameters, refer to the CNC operator's manual.

17) Function : **D_pitch_r** Reads pitch error data
Calling format : **short D_pitch_r (char ch_no, short start_no, short stop_no, uchar *file_name)**
Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
short start_no
Start correction position number
<numeric value>
(1 to 9999)
short stop_no
Stop correction position number
<numeric value>
(0 to 9999. When 0, omit this argument.)
uchar *file_name
Name of pitch error data file



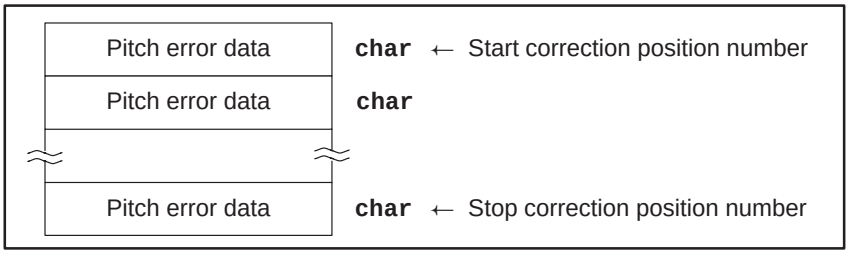
Return value : **= 0** Normal
= FFFF Communication error
= FFFE File open error
= FFFC File wite error
= FFF0 Sequence error
Others See Appendix F.

Outline : Directs the CNC to read pitch error data and sends the data from the start number to the stop number stored in the file. If the stop correction position number is omitted, only the data corresponding to the start correction position number is returned.

18) Function : **D_pitch_r_mem** Reads pitch error data.

Calling format :
```
short D_pitch_r_mem
(char ch_no, short start_no, short stop_no, long size, char *data)
```

Arguments :
char ch_no
Channel number (0 to 3) <numeric value>
short start_no
Start correction position number
<numeric value>
(1 to 9999)
short stop_no
Stop correction position number
<numeric value>
(0 to 9999. When 0, omit this argument.)
long size Buffer size <numeric value>
char *data Pitch error data buffer



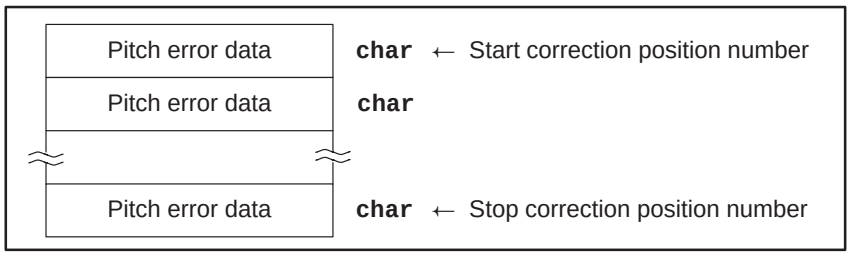
Return value : **= 0** Normal
= FFFF Communication error
= FFF0 Sequence error
Others See Appendix F.

Outline : Directs the CNC to read pitch error data and stores the data from the start number to the stop number in the buffer. If the stop correction position number is omitted, only the data corresponding to the start correction position number is returned.

19) Function : **D_pitch_w** Writes pitch error data

Calling format : **short D_pitch_w (char ch_no, short start_no, short stop_no, uchar *file_name)**

Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
short start_no
Start correction position number <numeric value>
(1 to 9999)
short stop_no
Stop correction position number <numeric value>
(1 to 9999)
uchar *file_name
Name of write pitch error data file



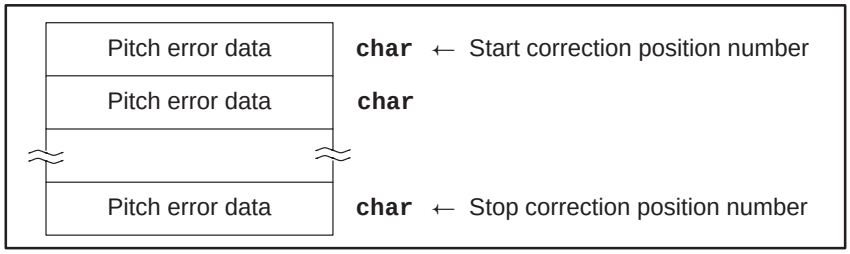
Return value : **= 0** Normal
= FFFF Communication error
= FFFE File open error
= FFFD File read error
= FFF0 Sequence error
Others See Appendix F.

Outline : Writes the pitch error data stored in the file from the start number to stop number to the CNC.

20) Function : **D_pitch_w_mem** Writes pitch error data.

Calling format : **short D_pitch_w_mem (char ch_no, short start_no, short stop_no, char *data)**

Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
short start_no
Start correction position number <numeric value>
(1 to 9999)
short stop_no
Stop correction position number <numeric value>
(1 to 9999)
char *data
Write pitch error data buffer



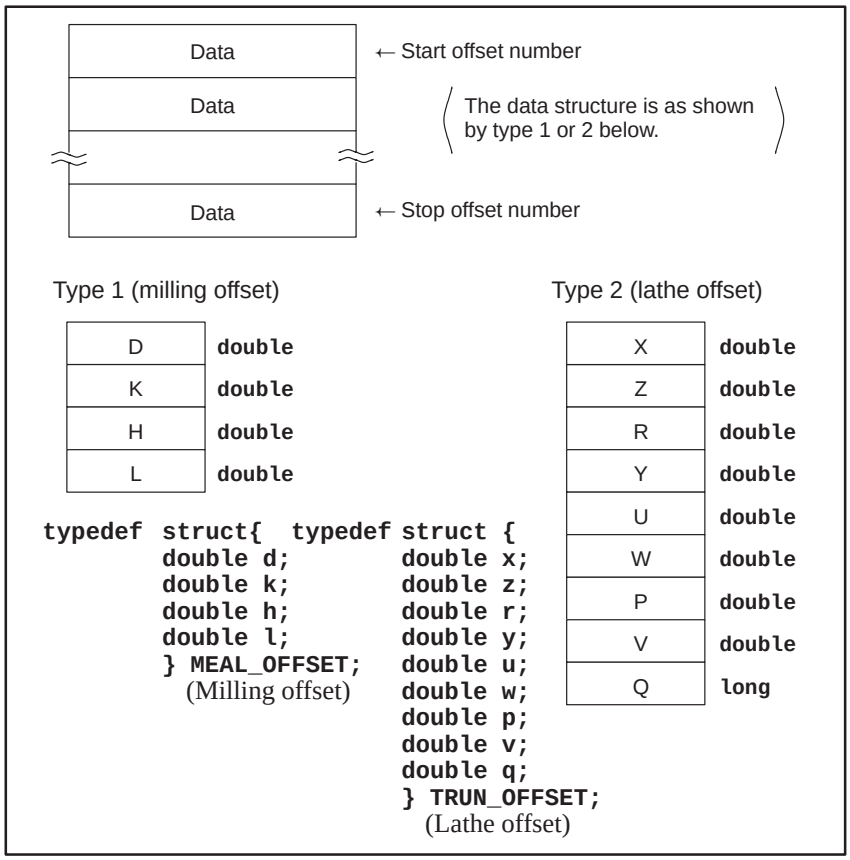
Return value : **= 0** Normal
= FFFF Communication error
= FFF0 Sequence error
Others See Appendix F.

Outline : Writes the pitch error data stored in the buffer from the start number to stop number to the CNC.

21) Function : **D_offset_r** Reads tool offset.

Calling format : **short D_offset_r (char ch_no, short start_no, short stop_no, ushort type, uchar *file_name)**

Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
short start_no
Start offset number (1 to 9999)
<numeric value>
short stop_no
Stop offset number <numeric value>
(0 to 9999. When 0, omit this argument.)
ushort type
Offset type specification <numeric value>
(Detailed information will be described later. Milling–type and lathe–type offsets cannot be specified together.)
For the setting conditions, see Table 4.2.4.15 in Section I–4.2.4.15 (5).
uchar *file_name
Data storage file name



Return value : **= 0** Normal
= FFFF Communication error
= FFFE File open error
= FFFC File write error
= FFF0 Sequence error
Others See Appendix F.

Outline : Directs the CNC to read tool offsets on a milling machine or lathe and stores the data from the start number to stop number in a file. If a stop number is omitted, only the data corresponding to the start number is returned.

Details of offset type
Bit

0	= Wear compensation	D : Cutter compensation
1	= Geometric compensation	K : Cutter compensation
2	= Wear compensation	X : X-axis offset quantity
3	= Wear compensation	Z : Z-axis offset quantity
4	= Wear compensation	R : Tool tip R offset
5	= Wear compensation	Y : Y-axis offset quantity
6	= Virtual tool tipQ	
7	= Not used	
8	= Wear compensation	H : Tool length offset
9	= Geomertic compensation	L : Tool length offset
10	= Geomertic compensation	U : X-axis offset quantity
11	= Geomertic compensation	W: Z-axis offset quantity
12	= Geomertic compensation	P : Tool tip R offset
13	= Geomertic compensation	V : Y-axis offset quantity
14	= Not used	
15	= Not used	

H, D, L, K : Milling machine offset
X, Z, R, Y, Q, U, W, P, V : Lathe offset

22) Function : **D_offset_r_mem** Reads tool offset.

Calling format :
```
short D_offset_r_mem
(char ch_no, short start_no, short
stop_no, ushort type, long size,
char *data)
```

Arguments :
char ch_no
Channel number (0 to 3) <numeric value>
short start_no
Start offset number (1 to 9999)
<numeric value>
short stop_no
Stop offset number <numeric value>
(0 to 9999. When 0, omit this argument.)
ushort type Offset type <numeric value>
(Milling type and lathe type cannot be specified at the same time.)
For the bit structure, see the tool offset read file.
long size Buffer size
char *data Data storage buffer.
(Memory size (in bytes) usable for the receive data storage buffer)
For the data structure, see the tool offset read file.

Return value : **= 0** Normal
= FFFF Communication error
= FFF0 Sequence error
Others See Appendix F.

Outline : Directs the CNC to read tool offsets on a milling machine or lathe and stores the data from the start number to stop number in a file. If a stop number is omitted, only the data corresponding to the start number is returned.

23) Function : **D_offset_w** Writes tool offset.

Calling format : **short D_offset_w (char ch_no, short start_no, short stop_no, ushort type, uchar *file_name)**

Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
short start_no
Start coffset number (1 to 9999)
<numeric value>
short stop_no
Stop offset number (1 to 9999)
<numeric value>
ushort type
Offset type specification <numeric value>
(Milling type and lathe type cannot be specified at the same time.)
For the bit structure, see the tool offset read file.
uchar *file_name
Write data storage file name.
Only items specified in the offset type are written.
For the data structure, see the tool offset read file.

NOTE
The offset type is specified for all write data. Be sure to input the data specified in type.

Return value : **= 0** Normal
= FFFF Communication error
= FFFE File open error
= FFFD File read error
= FFF0 Sequence error
Others See Appendix F.

Outline : Writes the offsets on milling machines or lathes stored in a file from the start number to stop number to the CNC.

24) Function : **D_offset_w_mem** Writes tool offset.

Calling format : **short D_offset_w_mem (char ch_no, short start_no, short stop_no, ushort type, char *data)**

Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
short start_no
Start offset number (1 to 9999)
<numeric value>
short stop_no
Stop offset number (1 to 9999)
<numeric value>
ushort type
Offset type specification <numeric value>
(Milling type and lathe type cannot be specified at the same time.)
For the bit structure, see the tool offset read file.
char *data Write data buffer.
Only items specified in the offset type are written.
For the data structure, see the tool offset read file.

NOTE
The offset type is specified for all write data. Be sure to input the data specified in type.

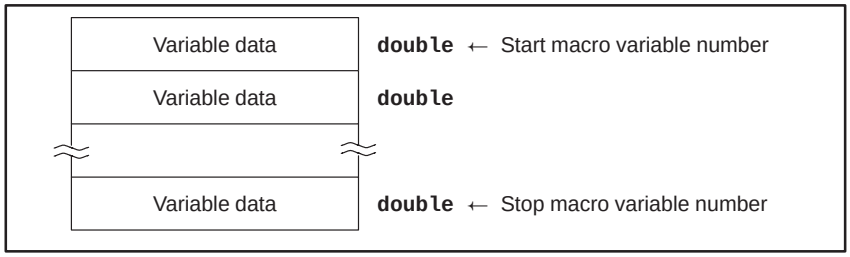
Return value : **= 0** Normal
= FFFF Communication error
= FFF0 Sequence error
Others See Appendix F.

Outline : Writes offsets on milling machines or lathes stored in a file from the start number to stop number to the CNC.

25) Function : **D_variable_r** Reads custom macro variables.

Calling format : **short D_variable_r (char ch_no, long start_no, long stop_no, uchar *file_name)**

Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
long start_no
Start macro variable number
<numeric value>
long stop_no
Stop macro variable number<numeric value>
(When 0, omit this operand.)
uchar *file_name
Macro variable data storage file name



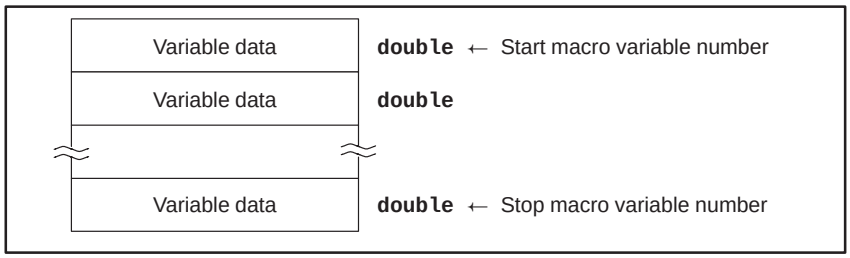
Return value : **= 0** Normal
= FFFF Communication error
= FFFE File open error
= FFFD File write error
= FFF0 Sequence error
Others See Appendix F.

Outline : Directs the CNC to read macro variables and stores the data from the start number to stop number in a file to return the data. If the stop number is omitted, only the data corresponding to the start number is returned.

26) Function : **D_variable_r_mem** Reads custom macro variables.

Calling format : **short D_variable_r_mem (char ch_no, long start_no, long stop_no, long size, double *data)**

Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
long start_no
Start macro variable number <numeric value>
long stop_no
Stop macro variable number<numeric value>
(When 0, omit this operand.)
long size Buffer size <numeric value>
double *data Macro variable data buffer



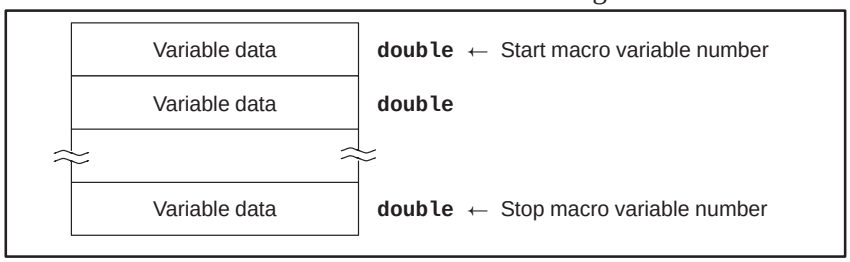
Return value : **= 0** Normal
= FFFF Communication error
= FFF0 Sequence error
Others See Appendix F.
In the case of “empty”, 0 is returned.

Outline : Directs the CNC to read macro variables and stores the data from the start number to stop number in buffer to return the data. If the stop number is omitted, only the data corresponding to the start number is returned.

27) Function : **D_variable_w** Writes custom macro variables.

Calling format : **short D_variable_w (char ch_no, long start_no, long stop_no, uchar *file_name)**

Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
long start_no
Start macro variable number
<numeric value>
long stop_no
Stop macro variable number<numeric value>
uchar *file_name
Macro variable data storage file name



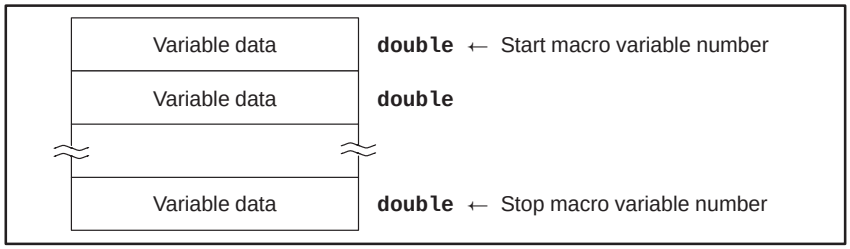
Return value : **= 0** Normal
= FFFF Communication error
= FFFE File open error
= FFFD File read error
= FFF0 Sequence error
Others See Appendix F.

Outline : Writes the macro variable data from the start number to stop number stored in a file to the CNC.

28) Function : **D_variable_w_mem** Writes custom macro variables.

Calling format : **short D_variable_w_mem (char ch_no, long start_no, long stop_no, double *data)**

Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
long start_no
Start macro variable number
<numeric value>
long stop_no
Stop macro variable number<numeric value>
double *data Macro variable data buffer



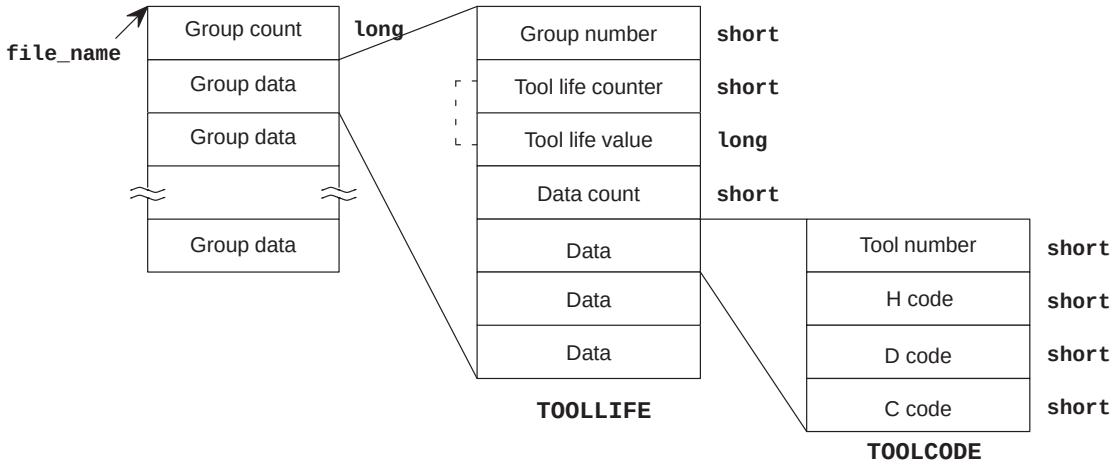
Return value : **= 0** Normal
= FFFF Communication error
= FFF0 Sequence error
Others See Appendix F.

Outline : Writes the macro variable data from the start number to stop number stored in the buffer to the CNC.

29) Function : **D_toollife** Reads tool life management data.

Calling format : **short D_toollife (char ch_no, short start_no, short stop_no, uchar *file_name)**

Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
short start_no
Start group number <numeric value>
short stop_no
Stop group number <numeric value>
(When 0, omit this operand.)
uchar *file_name
Tool life management data storage file name




```
typedef  struct{  typedef struct {
         short g_no;      short no;
         short type;      short hcode;
         long life;       short dcode;
         short d_cnt;     short ccode;
         } TOOLLIFE;      } TOOLCODE;
```

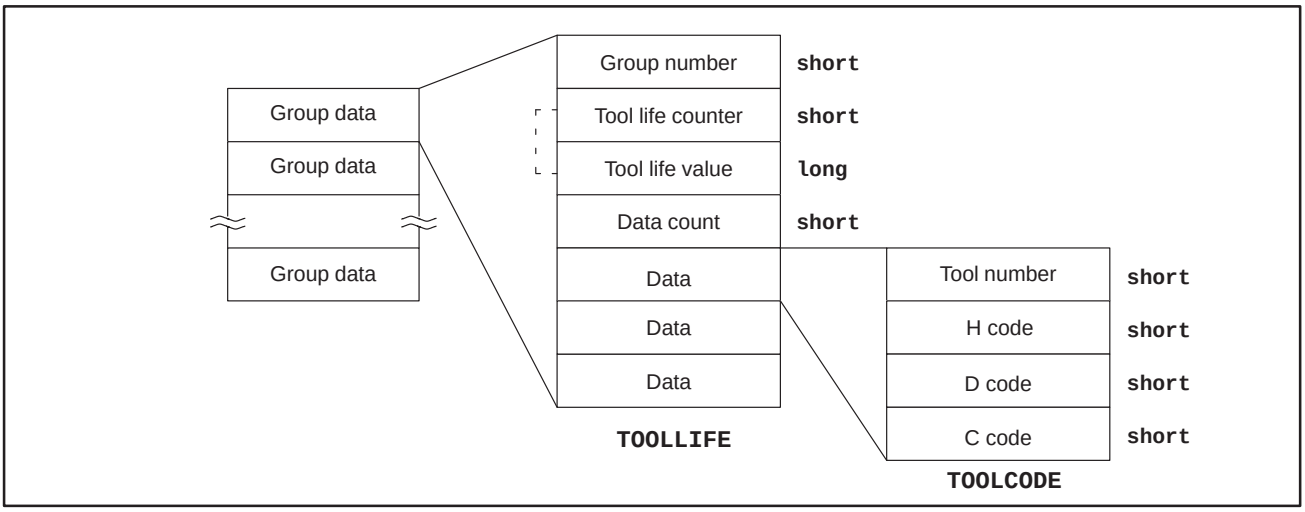
Return value : **= 0** Normal
= FFFF Communication error
= FFFE File open error
= FFFC File write error
= FFF0 Sequence error
Others See Appendix F.

Outline : Directs the CNC to read tool life management data and stores the data from the start number to stop number in a file. If the stop number is omitted, only the data corresponding to the start number is returned.

30) Function : **D_toollife_mem** Reads tool life management data.

Calling format : **short D_toollife_mem (char ch_no, short start_no, short stop_no, long size, long *count, TOOLLIFE *data)**

Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
short start_no
Start group number <numeric value>
short stop_no
Stop group number <numeric value>
(When 0, omit this operand.)
long size Buffer size <numeric value>
long *count
Read group data count storage address
TOOLLIFE *data
Tool life management data buffer



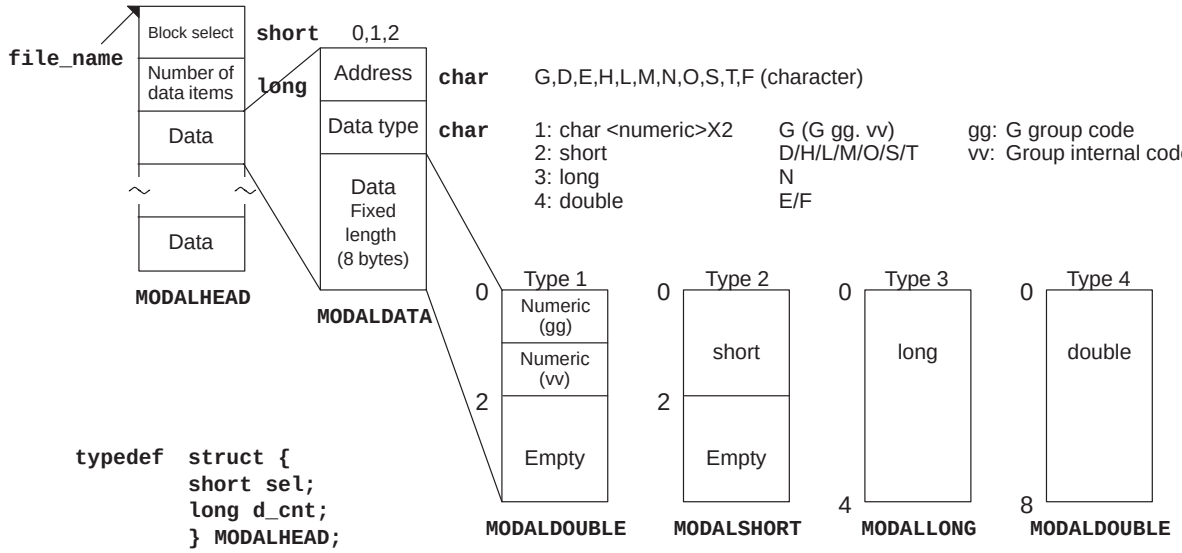
Return value : **= 0** Normal
= FFFF Communication error
= FFF0 Sequence error
Others See Appendix F.

Outline : Directs the CNC to read tool life management data, stores the data from the start number to stop number in buffer, and returns the data along with the number of group data items stored in the buffer. If the stop number is omitted, only the data corresponding to the start number is returned.

31) Function : **D_modal** Reads automatic operation modal information.

Calling format : **short D_modal (char ch_no, short select, uchar *file_name)**

Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
short select
0 [LAST_BLOCK] Preceding block <numeric value>
1 [ACTIVE_BLOCK] Current block
2 [BUFF_BLOCK] Next block
uchar *file_name
Modal data storage file name



Note) For an explanation of gg and vv, see Appendix A.8.

```
typedef  struct {
         short sel;
         long d_cnt;
         } MODALHEAD;
typedef  struct {
         char  addr;
         char  type;
         union  {
               MODLCHAR    c;
               MODALSHORT  s;
               MODALLONG   l;
               MODALDOUBLE d;

                } dat;
         } MODALDAT;
```

```
typedef  struct {
         char  dt1;
         char  dt2;
         short rsv1;
         long  rsv2;
         } MODALCHAR;

typedef  struct {
         short  dt;
         short  rsv1;
         long   rsv2;
         } MODALSHORT;
```

```
typedef  struct {
         long  dt;
         long  rsv;
         } MODALLONG;
```

```
typedef  struct {
         double  dt;
         } MODALDOUBLE;
```

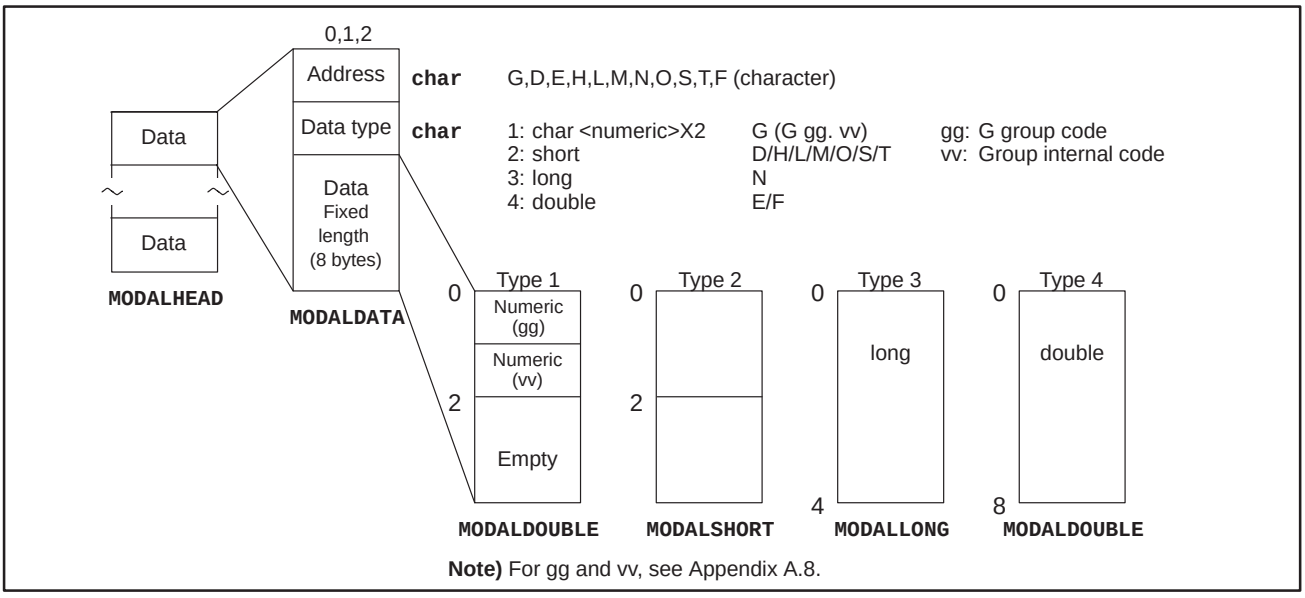
Return value : **= 0** Normal
= FFFF Communication error
= FFFE File open error
= FFFC File write error
= FFF0 Sequence error
Others See Appendix F.

Outline : Reads modal data and writes the read data to a file.

32)Function : **D_modal_mem** Reads automatic operation modal information.

Calling format : **short D_modal_mem (char ch_no, short select, long size,long *count, MODALDATA *data)**

Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
short select
0 [LAST_BLOCK] Preceding block
<numeric value>
1 [ACTIVE_BLOCK] Current block
2 [BUFF_BLOCK] Next block
long size Buffer size <numeric value>
long *count Read data count storage address
MODALDATA *data Modal data buffer



Note) For gg and vv, see Appendix A.8.

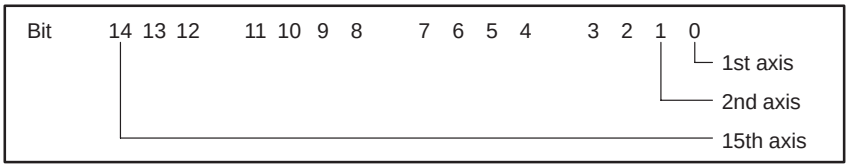
Return value : **= 0** Normal
= FFFF Communication error
= FFF0 Sequence error
Others See Appendix F.

Outline : Reads modal data, stores the read data in the buffer, and returns the data along with the number of data items stored.

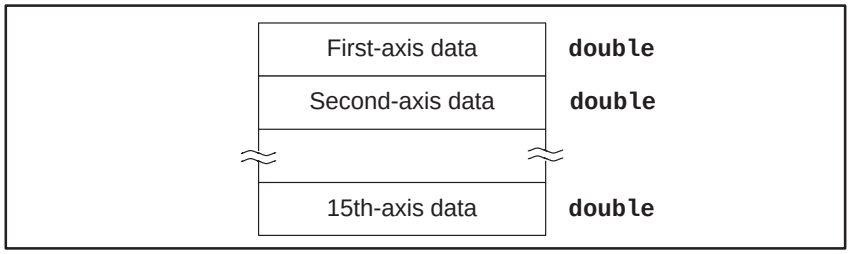
33)Function : **D_actualspeed** Reads actual speed.

Calling format : **short D_actualspeed (char ch_no, short axis, double *data)**

Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
short axis
Axis specification <numeric value>
(Turns on the bits corresponding to axes. Bits 0 to 14.)



double *data Actual speed data buffer



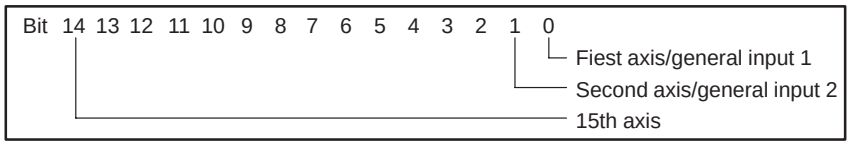
Return value : **= 0** Normal
= FFFF Communication error
= FFF0 Sequence error
Others See Appendix F.

Outline : Directs the CNC to read the actual speed and stores data of specified axes in buffer. Data of axes that are not specified is undefined.

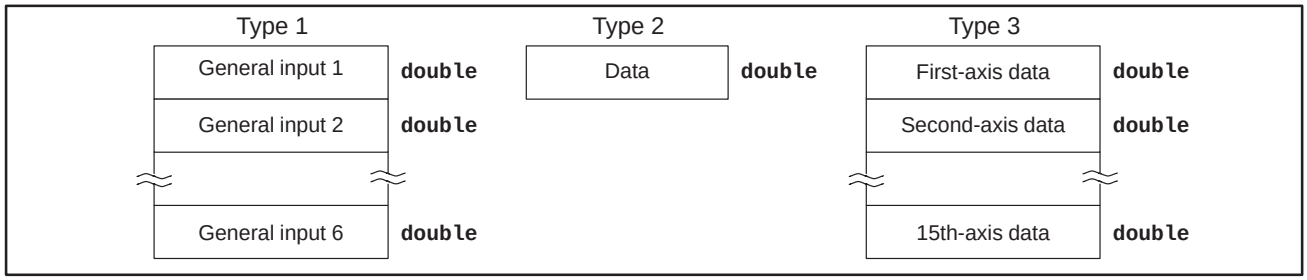
34) Function : **D_ADconvert** Reads A/D conversion data.

Calling format : **short D_ADconvert (char ch_no, short select, short axis,double *data)**

Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
short select
0 [G_PURPOSE] General input
(Axis is required: Bit 0 to 5)
100 [SPINDLE_1] Spindle 1 <numeric value>
101 [SPINDLE_2) Spindle 2
200 [AXIS_VOLTS] Axis data
(Axis is required: Bit 0 to 14)
short axis
Axis specification <numeric value>
(Turns on bits corresponding to axes or general input.)



double *data
A/D conversion data storage buffer.
Data types are types 1 to 3.



Return value : **= 0** Normal
= FFFF Communication error
= FFF0 Sequence error
Others See Appendix F.

Outline : Directs the CNC to read general input data, data for spindles 1 and 2, or axis data, and returns the data for each type. General input or axis data that is not specified is undefined.

35) Function : **D_alarm** Reads alarm information.

Calling format : **short D_alarm (char ch_no, ushort *alarm)**

Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
ushort *alarm
Alarm status storage area <numeric value>
Bit
0 – Background P.S. alarm
1 – Foreground P.S. alarm
2 – Overheat alarm
3 – Undefined
3 – Undefined
4 – Undefined
5 – SW alarm (parameters can be input)
6 – OT alarm (overtravel/spindle alarm)
7 – PMC error
8 – External alarm
9 – Undefined
10 – Serious P.S. alarm
11 – Undefined
12 – Servo alarm
13 – I/O alarm
14 – PW alarm
15 – Undefined

Return value : **= 0** Normal
= FFFF Communication error
= FFF0 Sequence error
Others See Appendix F.

Outline : Directs the CNC to read alarm information and returns the information with bit patterns.

36) Function : **D_status** Reads status information.

Calling format : **short D_status (char ch_no, ushort *status, ushort *alarm)**

Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
ushort *status
CNC status storage area <numeric value>
Bit
0 – RWD rewind
1 – AL alarm
2 – RST reset
3 – SPL automatic operation stopped
4 – STL automatic operation activated
5 – OP automatic operation
6 – SA servo setup complete
7 – MA CNC setup complete
8 – Not used
9 – Not used
10 – Not used
11 – Not used
12 – M00 : M00 output
13 – M01 : M01 output
14 – M02 : M02 output
15 – M30 : M30 output
ushort *alarm
Alarm status storage area <numeric value>
(For details, see the alarm information read function.)

Return value : **= 0** Normal
= FFFF Communication error
= FFF0 Sequence error
Others See Appendix F.

Outline : Directs the CNC to read status information and returns the status information in the form of a bit pattern. When the alarm bit in the status information is on, returns alarm information at the same time.

37) Function : **D_message** Message output

Calling format : **short D_message (char ch_no, char buf_no, uchar *message)**

Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
char buf_no
Message buffer number <numeric value>
uchar *message
Output message <character string>
(A maximum of 32 characters. Terminate the text with a null code.)

Return value : **= 0** Normal
= FFFF Communication error
= FFF0 Sequence error
Others See Appendix F.

Outline : Outputs a message to the CNC.

38) Function : **D_PMCdata_r** Reads PMC data.

Calling format : **short D_PMCdata_r (char ch_no, long addr, short length, short status, uchar *data)**

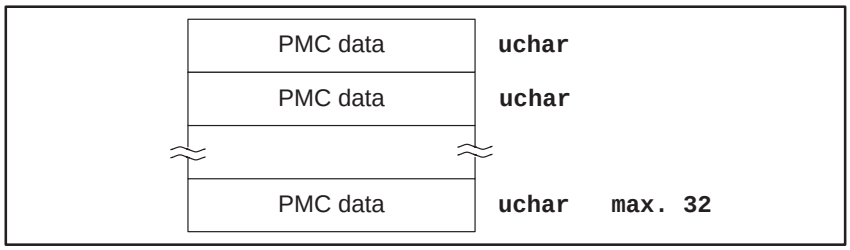
Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
long addr
PMC RAM address or data number
<numeric value>

–PMC RAM address (Series 15–A and address specification type of Series 15–B)
The address is specified with eight hexadecimal characters by prefixing 0X41 to the 6–digit physical address, given in the list of relationships between PMC addresses and physical addresses in Section 3.3.2 of the “FANUC PMC–MODEL N Programming Manual” (B–61013–2).
Example : For F000, specify 0X41FFE200.

–Data number (Series 0)
Refer to the “FANUC Series 0 Connection Manual”(B-61393) and “FANUC PMC MODEL-K/L/M Programming Manual”(B–55193).
Specify the numeric part of the PMC address, which is the PMC address less the initial alphabetic character.
Example : For F148, specify 148.
For R700, specify 700.

–Data number (Data number specification type of Series 15–B and Series 16/18/21)
See Section 4.2.4.26.
Specify the numeric value corresponding to the first alphabetic character of a PMC address, plus a 4–digit decimal number.
Example : For R123, specify 60123.

short length
Data length (1 to 32) <numeric value>
short status
1 : RAM address, 2 : Data number
<numeric value>
uchar *data PMC data buffer



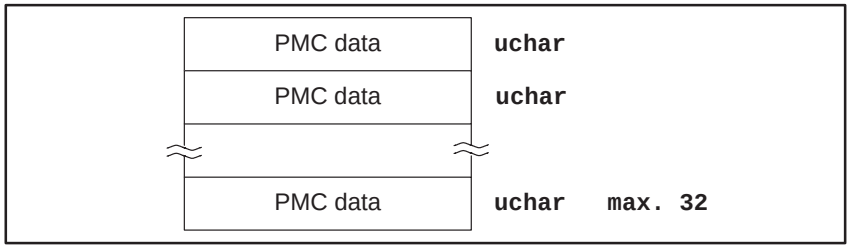
Return value : **= 0** Normal
= FFFF Communication error
= FFF0 Sequence error
Others See Appendix F.

Outline : Directs the CNC to read PMC data and returns as much PMC data as specified by the data length, beginning at the specified address.
This function is generally used for handshaking with the PMC. The CNC status can be obtained by reading the DO signal of the CNC.

39) Function : **D_PMCdata_w** Writes PMC data.

Calling format : **short D_PMCdata_w (char ch_no, long addr, short length, short status, uchar *data)**

Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
long addr
PMC RAM address or data number
<numeric value>
For an explanation of how to specify the RAM address and data number, see the description of the D_PMCdata_r function in 38).
short length
Data length (1 to 32) <numeric value>
short status
1 : RAM address, 2 : Data number
<numeric value>
uchar *data Write PMC data buffer



Return value : **= 0** Normal
= FFFF Communication error
= FFF0 Sequence error
Others See Appendix F.

Outline : Writes as much PMC data as specified by the data length beginning at the specified address to the CNC.
This function is generally used to give directions to the PMC. This function permits handshaking with the PMC in combination with the PMC data read function.

40) Function : **D_select** Searches for a program number.

Calling format :
```
short D_select
(char ch_no, char *prg_no)
```

Arguments :
char ch_no
Channel number (0 to 3) <numeric value>
char *prg_no
Program number searched for
<character string>
(o0000 to o9999)

Return value :
= 0 Normal
= FFFF Communication error
= FFF0 Sequence error
Others See Appendix F.

Outline : Directs the CNC to search for a program number. In the tape mode, this function specifies the program number for performing a DNC operation.

* Release background edit mode and simultaneous edit mode (Series 15 only) in advance.
* With the Series 15, the sequence terminates normally in any mode, but the function is made effective only in edit mode and memory mode.

41) Function : **D_start** Cycle start

Calling format :
```
short D_start
(char ch_no, char *prg_no)
```

Arguments :
char ch_no
Channel number (0 to 3) <numeric value>
char *prg_no
Program number to be started
<character string>
(o0000 to o9999)
If o0000 or a null code is entered, the program already selected is started.

Return value :
= 0 Normal
= FFFF Communication error
= FFF0 Sequence error
Others See Appendix F.

Outline : Specifies a program number for the CNC and starts the program. If the program number is omitted, the CNC starts the executable program specified in the program number search function.
If a complex condition check is required at cycle start, avoid starting a program using this function. Use the PMC data write function instead.

42) Function : **D_reset** Reset
Calling format : **short D_reset (char ch_no)**
Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
Return value : **= 0** Normal
= FFFF Communication error
= FFF0 Sequence error
Others See Appendix F.
Outline : Directs the CNC to reset the system.

43) Function : **D_systemID** Reads system ID.
Calling format : **short D_system ID (char ch_no, uchar *name, uchar *revision)**
Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
uchar *name
CNC model name <character string>
(A maximum of 32 characters. Terminate the model name with a null code.)
uchar *revision Version <character string>
(A maximum of eight characters. Terminate the version with a null code.)
Return value : **= 0** Normal
= FFFF Communication error
= FFF0 Sequence error
Others See Appendix F.
Outline : Directs the CNC to read the system ID. Returns the CNC model name and DNC2 software version. This function indicates the type of CNC connected.

44) Function : **D_initial** Initializes communication port.
Calling format : **short D_initial (char ch_no)**
Arguments : **char ch_no**
Channel number (0 to 3) <numeric value>
(sys_table)
Initialization parameter
Return value : **= 0** Normal
= FFFF Initialization error
= FFFE Parameter error
Outline : Initializes the specified communication port.

5 EXAMPLE OF USING THE COMMUNICATION PROGRAM

This section describes a simple example in which the functions in the communication program library are used, using list 5–1.

(1) Include the required files (5 and 6).
- Enclose the include files required in the standard library of the compiler in < > and include files of this library in double quotation marks.

(2) Initialize the communication port (14 to 20).
- Communication parameters can be reset to values different from the initial values as shown in lines 14 and 15. (For initial values and reset values, see Appendix B.1.)
- Call the initialization library and store the result in ans. (Line 16)
- When the result of initialization is not normal, output a message (line 18) and terminate the program. (Line 19)

(3) Download the NC program (22 to 34).
- Call the download library and store the result in ans. (Line 22) Download the program with port number 0, download file name “pr_file.dat”, and program number 0001.
- Display a message according to the result. (Lines 23 to 34)

LIST 5–1

```
 1 :  /***************************************/
 2 :  /* Example of coding */
 3 :  /***************************************/
 4 :
 5 :  #include<stdio.h>
 6 :  #include”dnc2def.h”
 7 :
 8 :  main( );
10 :  main( )
11 :  {
12 :  short ans;
13 :
14 :     sys_table.port [0].speed = 4;        /* Set communication speed to 9600 bps */
15 :     sys–table.port [0].parity = 1;       /* No parity bit is set */
16 :     ans = D_initial (0);                 /* Initialize port 0 */
17 :     if (ans ! = 0) {                        /* Is initialization successful? */
18 :        printf (”Initialization error \n”)   /* Initialization error */
19 :        exit(l);                             /* Return to DOS */
20 :     }
21 :
22 :     ans = D_download (0, ”pr_file.dat”, ”O0001”);/*Download */
23 :     if (ans == 0) {                         /* Check the result */
24 :        printf (”Download termination \n”);  /* Normal termination */
25 :     } else if (ans == 0xfffe) {
26 :        printf (”No file is found \n”)       /* File not found */
27 :        exit(l);
28 :  } else if (ans == 0xfffd) {
29 :        printf (”The file cannot be read \n”)/* File read error */
30 :        exit(l);
31 :     } else {
32 :        printf (”Download failure \n”)       /* Communication error, other errors */
33 :        exit(l);
34 :     }
35 :     return 0;
36 :  }
```

6 ERROR CODES

6.1 VALUES RETURNED BY FUNCTIONS

The return values 0FFFFh to 0FFCFh (absolute values of 1 to 49) output from functions in this library indicate errors detected within the library. The return values of 0FFCEh or less (absolute values of 50 or greater) indicate errors output when a negative response is sent from the CNC. For details of the error codes of this library, see Appendix J.

6.2 COMMUNICATION ERROR INFORMATION

If values returned from functions indicate communication error (0FFFFh), the details of the cause are stored in the variable dnc_link_error. Refer to it an necessary.
The following values are returned to short dnc link error from the functions:

`short dnc_link_error`

1 : Time–out
2 : Retry over
3 : Busy (during text transmission, a response to ENQ was NAK)
4 : Protocol error (when receiving text, other data was received when waiting for ENQ)
5 : Warning (when receiving text, a time–out occurred when waiting for EOT. However, text is normally received.)
11 : Line disconnected
12 : Overrun error
13 : Framing error
14 : Parity error
21 : ISO code parity error

The communication error having code 1 occurs when the receive–disabled state occurs for each retry within the time–out period set for receiving or sending each character. As soon as this error is detected, processing is discontinued.

This does not apply while waiting for a down–load request (**`D_download_wait`**) or while waiting for a command from the CNC. In these cases the system keeps waiting for a command even though a time–out occurs.

The communication error having code 2 occurs when a retry is made in the data link layer level for any cause other than time–out and the defined number of retries is repeated. As soon as this error is detected, processing is stopped.

The communication errors having codes 3 to 9 are data link layer protocol errors. As soon as these errors occur, processing is stopped.

The communication errors having codes 10 or more are communication line errors. As soon as these errors are detected, processing is stopped.

7 ERROR LOG INFORMATION

When an error occurs in this library, the error, the time it occurred, and other information can be saved in a file. A function called the logging function, is provided for saving this information.

There are the following four types of logs depending on the cause of the error:sequence error log, protocol error log, phase trace log, and communication error log.

To use the logging function, set the appropriate log ON/OFF switch of system parameter to 1 : ON.

This logging function is used to check programs. Normally, do not use it. Since phase trace logs are taken at all times, the file input–output time increases, affecting the time–out period. Therefore, use the logging function carefully.

7.1 DETAILS OF THE ERROR LOG FILE

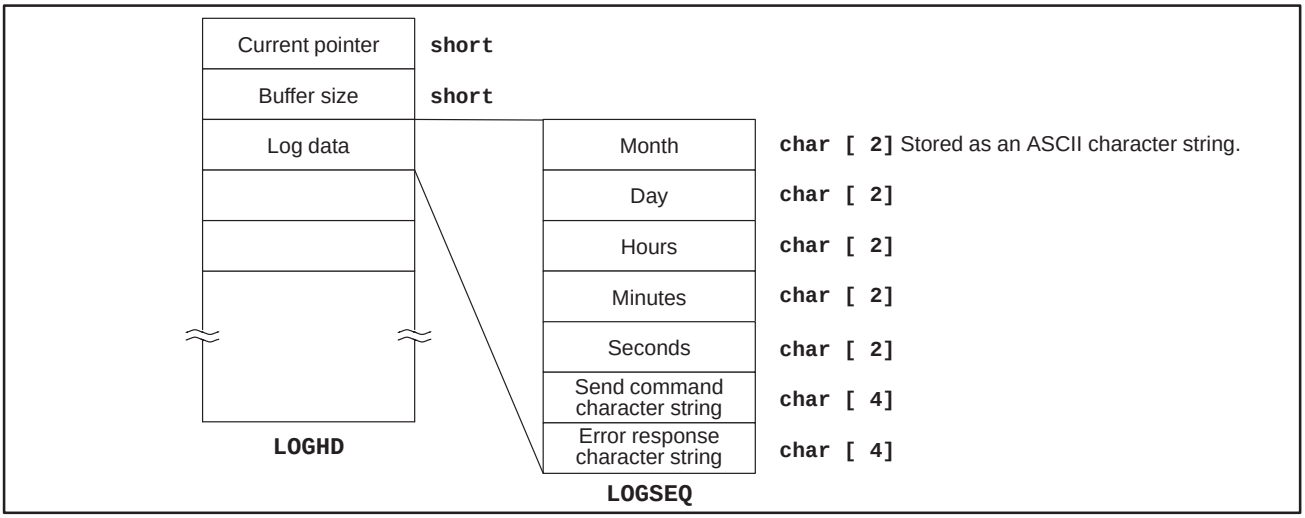
(1) Sequence error log

When a command is sent to the NC and a response to the command is not the normal sequence, the sequence data is logged.

The log file name is <**sequence.log**> and its contents are as shown below.

When the appropriate log switch of the system parameter table is not ON, logging is not performed.

Sequence error log file (file name **: sequence.log**)



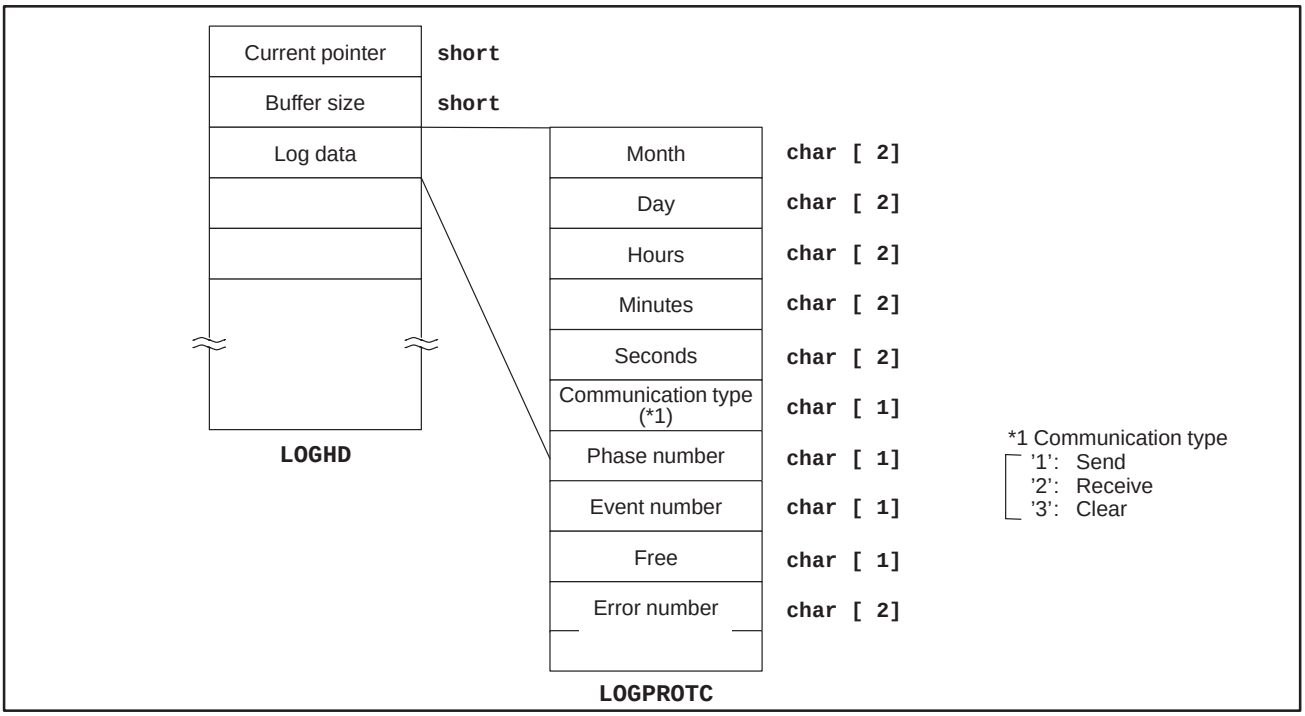
(2) Protocol error log

In the data link layer, when protocol does not terminate normally during text send/receive, the error is logged.

The log file name is <**protocol.log**> and its contents are as shown below.

When the appropriate log switch of the system parameter table is not ON, logging is not performed.

Protocol error log file (file name **: protocol.log**)



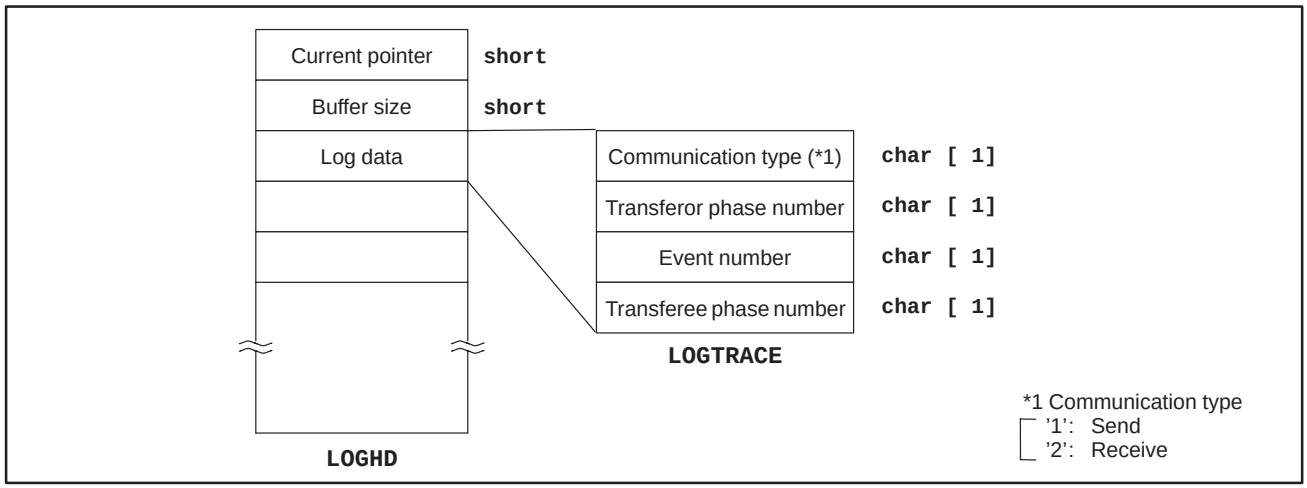
(3) Phase trace log

Phase transition is traced during text send/receive in the data link layer and the phase number is logged.

The log file name is <**phtrace.log**> and its contents are as shown below.

When the appropriate log switch of the system parameter table is not ON, logging is not performed.

Phase trace log file (file name **: phtrace.log**)



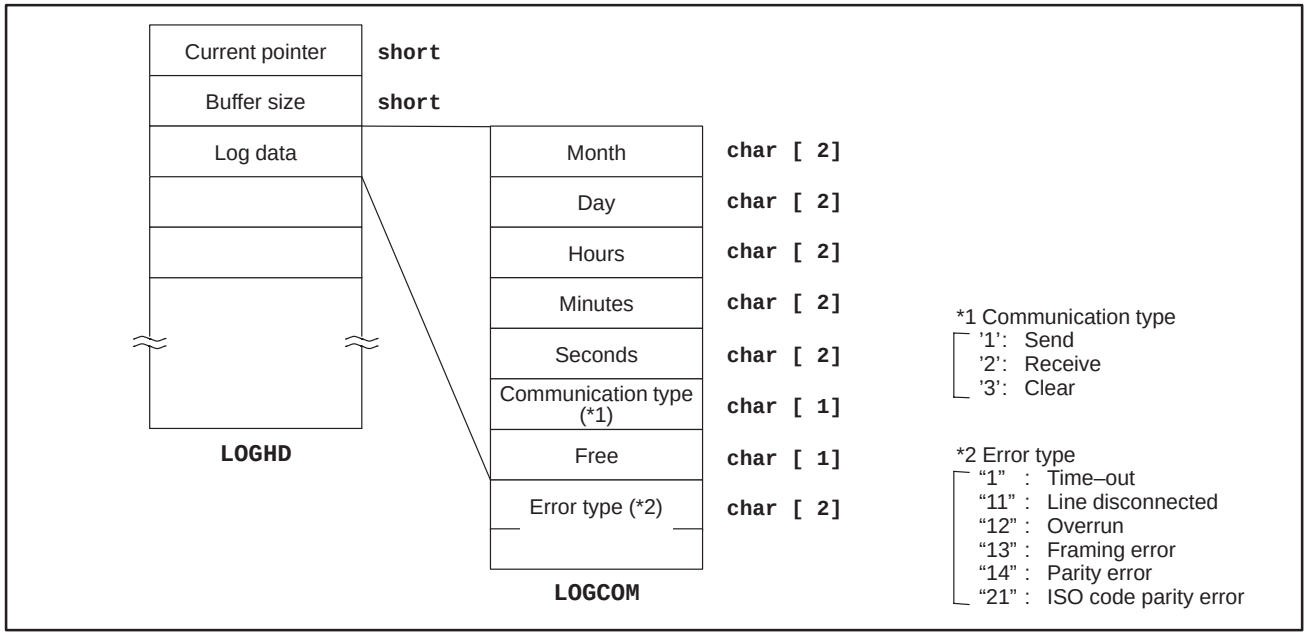
(4) Communication error log

When an error such as an overrun error occurs while sending or receiving one character, or when a send/receive time–out occurs, the error is logged.

The log file name is <**comerror.log**> and its contents are as shown below.

When the appropriate log switch of the system parameter table is not ON, logging is not performed.

Communication error log file (file name **: comerror.log**)



7.2 SIMPLE ERROR LOG BROWSING FUNCTION

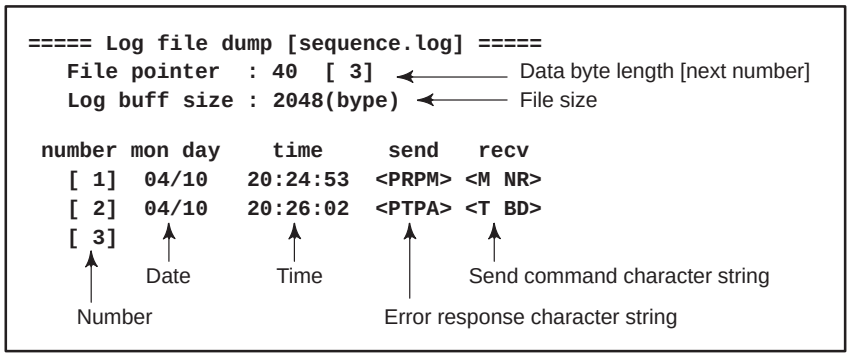
The command <**log.exe**> simplifies the browsing of the contents of each log file.

A simple help message is displayed by entering "**C>log -?.**"

The method for browsing each log file is described below.

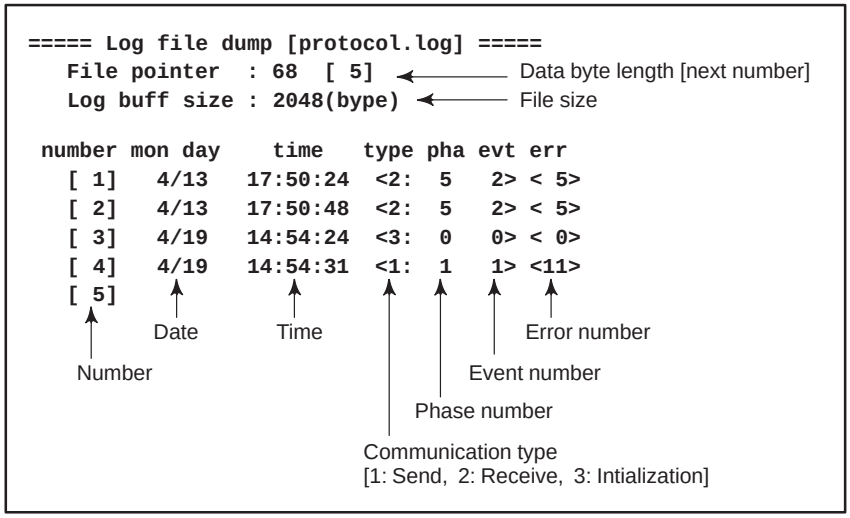
(1) Sequence error log

The information shown below is displayed by entering "**C>log -seq**", enabling sequence error logs to be browsed.



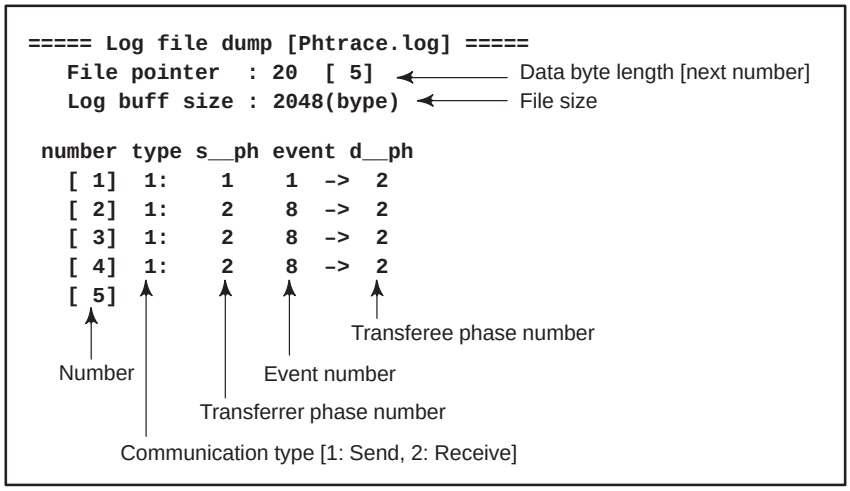
(2) Protocol error log

The information shown below is displayed by entering "**C>log -pro**", enabling protocol error logs to be browsed.



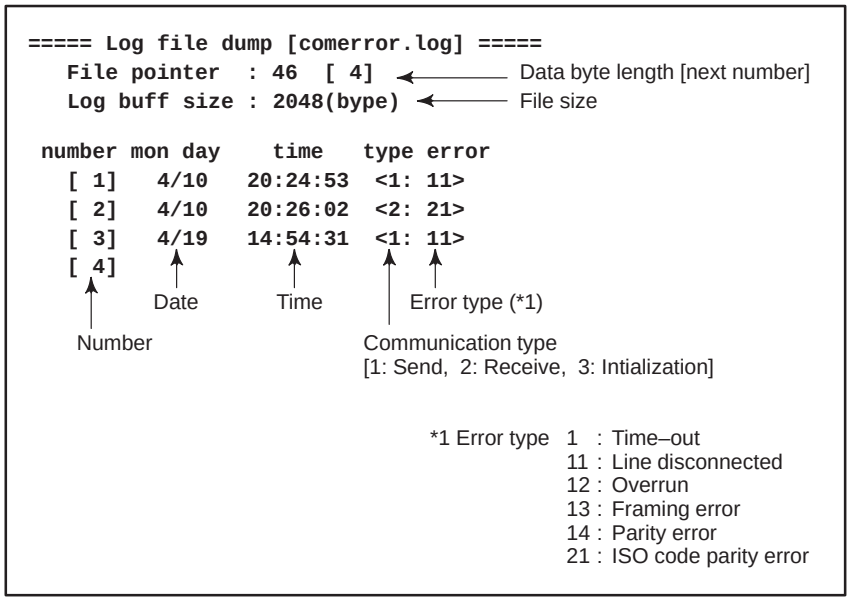
(3) Phase trace log

The information shown below is displayed by entering "**C>log –pha**", enabling phase trace logs to be browsed.



(4) Communication error log

The information shown below is displayed by entering "**C>log –com**", enabling communication error logs to be browsed.

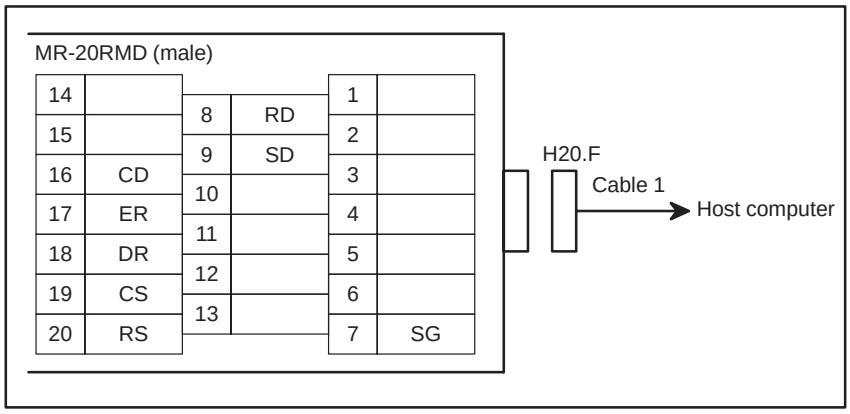


APPENDIX

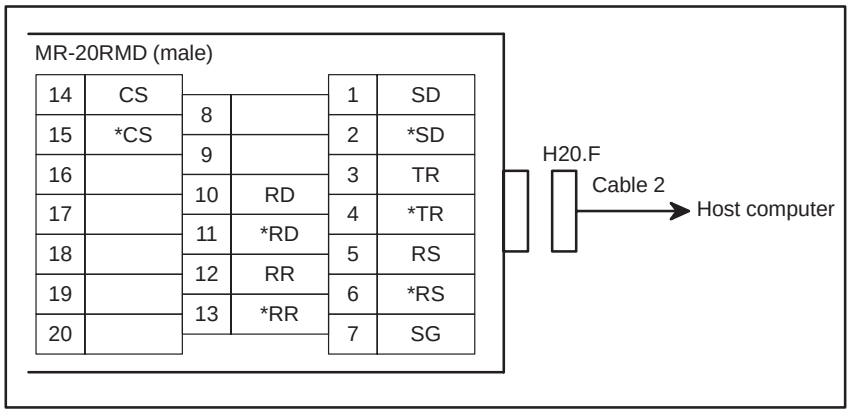
A CONNECTOR SPECIFICATIONS

A.1 FANUC Series 0 (A02B–0098–J055)

(1) RS-232-C
M77 (in the CNC cabinet)



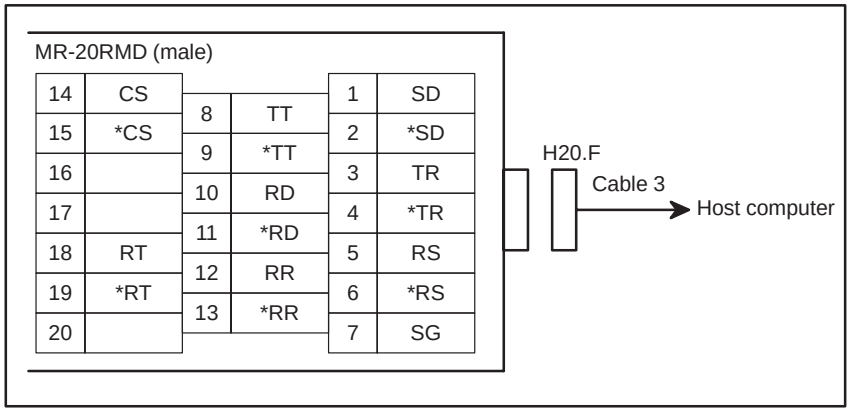
(2) RS-422
(a) M77 (in the CNC cabinet)



If a synchronization reception clock is not necessary, the M77 can also be used for the RS-422 interface.

To use the M77 for the RS-422 interface, do not wire TT (∗TT) and RT (∗RT).

(b) M73 (in the CNC cabinet)



A.2 FANUC Series 15–A

(1) RS-232-C
CD4 (in the CNC cabinet)

MR-20RF (female)

14	+24V	8	RD	1	
15		9	SD	2	
16	CD	10		3	
17	SG	11		4	
18	DR	12		5	ER
19	CS	13		6	
20	RS			7	

(2) RS-422
CD3 (in the CNC cabinet)

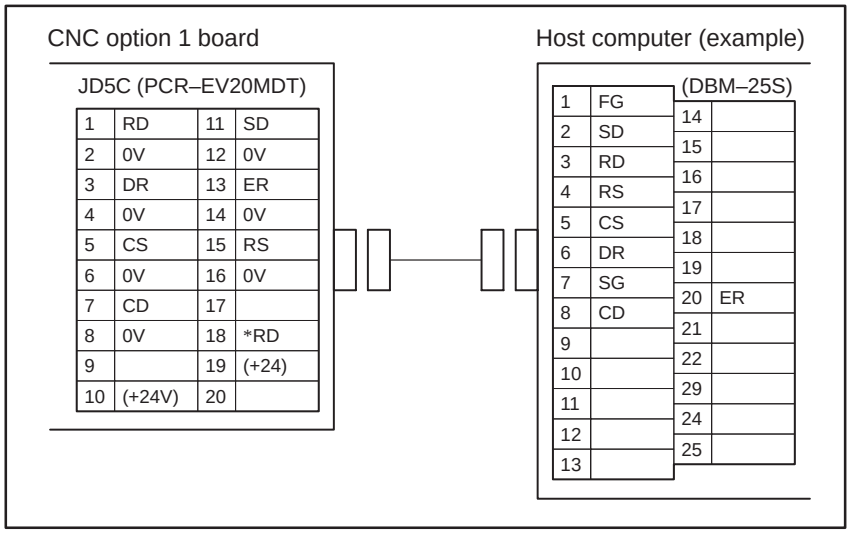
MR-20RM (male)

14	CS	8	TT	1	SD
15	*CS	9	*TT	2	*SD
16		10	RD	3	TR
17		11	*RD	4	*TR
18	RT	12	RR	5	RS
19	*RT	13	*RR	6	*RS
20				7	SG

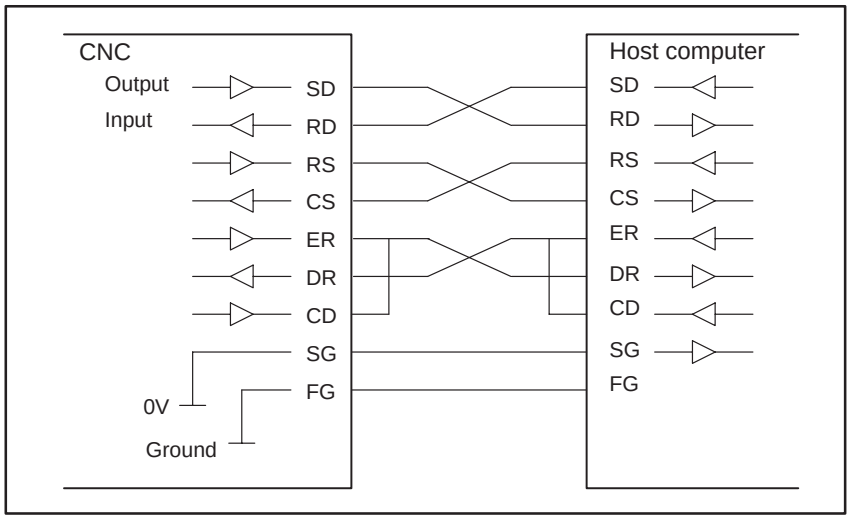
A.3 FANUC Series 15–B AND Series 16/18

(1) RS–232C

RS–232C connection signal diagram



Connection between RS–232C equipment

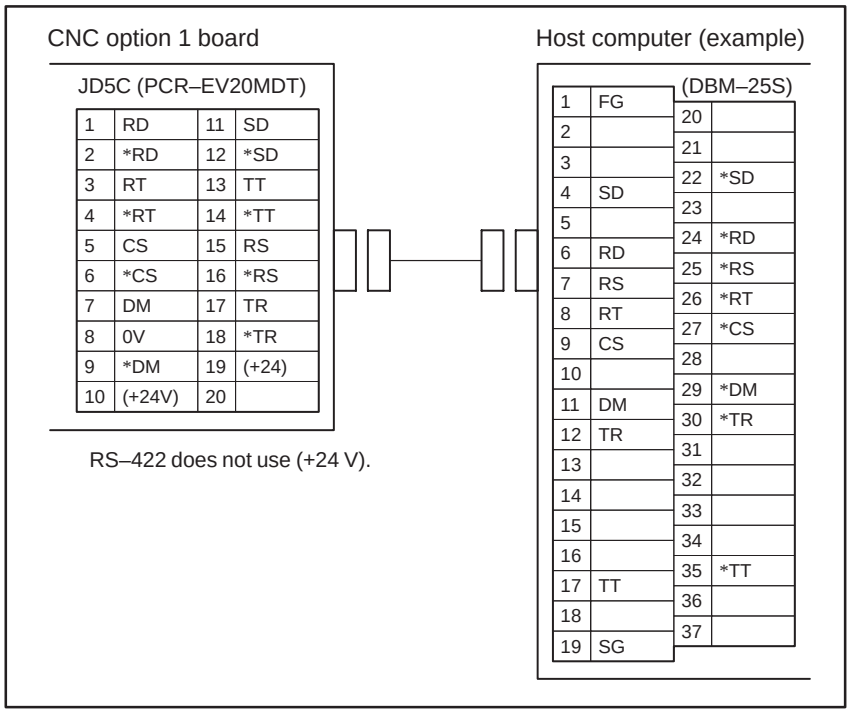


NOTE

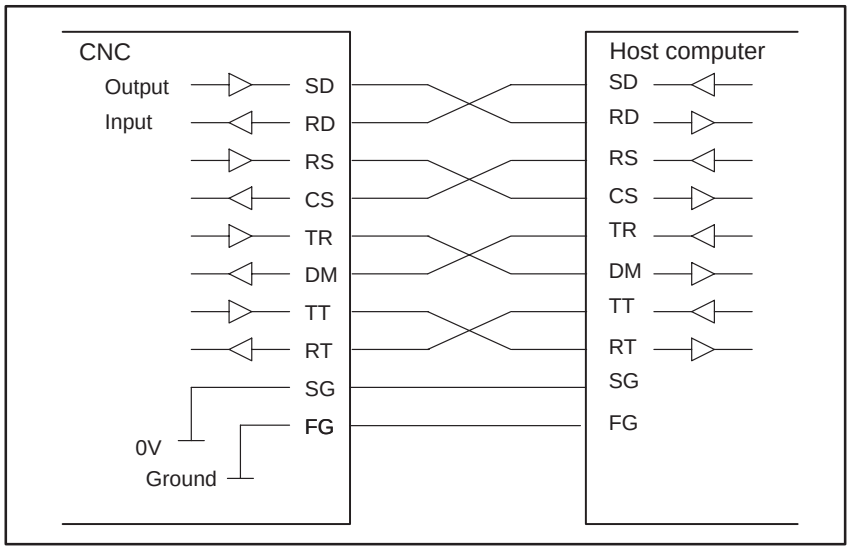
When CS is not used, connect CS and RS. When the IBM PC–AT is used as the host computer, connect CS and RS in the CNC. When DR is not used, connect DR and ER. Always connect CD and ER.

(2) RS–422

RS–422 connection signal diagram



Connection between RS–422 equipment



B PARAMETERS

The following parameters are sent to the DNC2 system only when the CNC is turned on. After changing any of the parameters, turn off and on again the power switch of the CNC.

B.1
FANUC Series 0 (UPPER ROW: Series 0–TC, LOWER ROW: Series 0–MC)

	#7	#6	#5	#4	#3	#2	#1	#0
0051			ECLK	NCKCD	RSASCI		PARTY	STP2
			ECLK	NCKCD	RSASCI		PARTY	STP2

ECLK : Reception clock source (for RS-422 only)
0 : Internal
1 : External

When ECLK is on, the M77 connector is used.

NCKCD : Whether to check CD (signal quality detection) in the RS-232-C
0 : Check
1 : No check

RSASCI : Transmission code
0 : ISO code
1 : ASCII code

> **NOTE**
> With the Series 0, use ASCII code.

PARTY : Parity
0 : No parity check
1 : Even parity

STP2 : Number of stop bits
0 : 1
1 : 2

> **NOTE**
> The data length (in bits) is determined according to what is set in RSASCI and PARTY. If both RSASCI and PARTY are 1, the data is 7 bits long. Otherwise, it is 8 bits long.

	#7	#6	#5	#4	#3	#2	#1	#0
0055					RS42			
					RS42			

RS42 : Interface used between the host and remote buffers
0 : RS-232-C interface
1 : RS-422 interface

0251 0251	Baud rate

Range of data : 1 to 12 (RS-232-C) or 5 to 15 (RS-422)
If a value out of the valid range is specified, either the maximum or minimum value is assumed. The default value is 10 (4800 bps).

Value	Baud rate	Value	Baud rate	Value	Baud rate
1	50	6	300	11	9600
2	100	7	600	12	19200
3	110	8	1200	13	38400
4	150	9	2400	14	76800
5	200	10	4800	15	86400

0365 0320	Timeout value for the no-response timer

Range of data : 1 to 60 (seconds)
If a value out of the valid range is specified, either 1 or 60 is assumed. The default value is 5 (seconds).

0366 0321	Timeout value for the EOT signal timer

Range of data : 1 to 60 (seconds)
If a value out of the valid range is specified, either 1 or 60 is assumed. The default value is 5 (seconds).

0368 0323	Maximum number of times that a prompt is made regarding an invalid transmission protocol or no-response in the data link layer

Range of data : 1 to 10 (times)
If a value out of the valid range is specified, either 1 or 10 is assumed. The default value is 5 (times).

0369 0324	Maximum number of times that a message is retransmitted when not acknowledged.

Range of data : 1 to 10 (times)
If a value out of the valid range is specified, either 1 or 10 is assumed. The default value is 3 (times).

0370 0325	Maximum number of characters that can be received after transmission has been stopped (CS off)

Range of data : 10 to 256 (characters)
If a value out of the valid range is specified, either 10 or 256 is assumed. The default is 256 characters. Since the maximum value that can be set in this parameter is 255, when 256 is to be specified, set 0.

	#7	#6	#5	#4	#3	#2	#1	#0
0396					ERCODE	NCKER		NCKBCC
					ERCODE	NCKER		NCKBCC

NCKBCC : Whether to check the BCC value
0 : Check
1 : No check; the BCC cannot be omitted.

NCKER : Whether to check the ER (RS-232-C)/TR (RS-422) signal
0 : Check
1 : No check

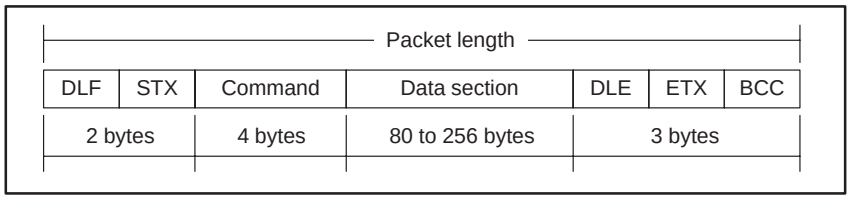
ERCODE : Error code for negative response
1 : Negative response is indicated by an error code consisting of four hexadecimal digits.

0 : Negative response is not indicated by an error code.

When using the FANUC-created DNC2 communication program library in the host computer, set this parameter to 1.

0490 0490	Maximum datagram (data section) length

Range of data : 80 to 256 (characters)
If a value out of the valid range is specified, either 80 or 256 is assumed. The default value is 256 (characters).
This parameter defines a maximum packet length for DNC2 transmission. The maximum length of a packet is the length specified by parameter 0490 plus nine characters including the first two characters, four command characters, and the last three characters.



B.2
Series 15

	#7	#6	#5	#4	#3	#2	#1	#0
0000						ISP		

(Power must be turned off and on again.)

ISP : Transmission code
0 : ISO code
1 : ASCII code

0020	Interface number for the foreground input device
0021	Interface number for the foreground output device
0022	Interface number for the background input device
0023	Interface number for the background output device

To use DNC2, set this parameter to 4.

	#7	#6	#5	#4	#3	#2	#1	#0
5000				NCKCD			PARTY	RB422

(Power must be turned off and on again.)

RB422 : Interface used between the host and remote buffers
0 : RS-232-C
1 : RS-422

PARTY : Parity
0 : Even parity
1 : No parity check

NCKCD : Whether to check CD (signal quality detection) in the RS-232-C
0 : Check
1 : No check

NOTE
The data length (in bits) is determined according to what is set in RSASCI and PARTY. If both RSASCI and PARTY are 1, the data is 7 bits long. Otherwise, it is 8 bits long.

5070	Minimum baud rate that causes the reception lock pulse to be supplied from the host

(Power must be turned off and on again.)

Range of data : 5 to 15
If the baud rate is grater than or equal to the value specified in this parameter, the reception clock pulse is supplied from the host. The parameter is valid for the RS-422 interface only.

5072	RS-422 stop bit
5082	RS-232-C stop bit

(Power must be turned off and on again.)

Range of data : 1 to 2
1 : 1 stop bit
2 : 2 stop bits
If the value set in the parameter is not 2, 1 is assumed.

5073	RS-422 baud rate
5083	RS-232-C baud rate

(Power must be turned off and on again.)

Range of data : 1 to 12 (RS-232-C) or 5 to 15 (RS-422)
If a value out of the valid range is specified, either the maximum or minimum value is assumed. The default value is 10 (4800 bps).

Value	Baud rate	Value	Baud rate	Value	Baud rate
1	50	6	300	11	9600
2	100	7	600	12	19200
3	110	8	1200	13	38400
4	150	9	2400	14	76800
5	200	10	4800	15	86400

5050	Timeout value for the no-response timer

Range of data : 1 to 60 (seconds)
If a value out of the valid range is specified, either 1 or 60 is assumed. The default value is 5 (seconds).

5051	Timeout value for the EOT signal timer

Range of data : 1 to 60 (seconds)
If a value out of the valid range is specified, either 1 or 60 is assumed. The default value is 5 (seconds).

5052	Maximum number of times that a prompt is made regarding an invalid transmission protocol or no-response in the data link layer

Range of data : 1 to 10 (times)
If a value out of the valid range is specified, either 1 or 10 is assumed. The default value is 5 (times).

5053	Maximum number of times that a message is retransmitted when not acknowledged.

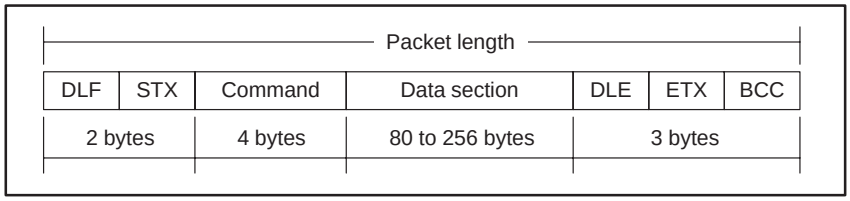
Range of data : 1 to 10 (times)
If a value out of the valid range is specified, either 1 or 10 is assumed. The default value is 3 (times).

5054	Maximum number of characters that can be received after transmission has been stopped (CS off)

Range of data : 10 to 256 (characters)
If a value out of the valid range is specified, either 10 or 256 is assumed.
The default value is 256 (characters).

5055	Maximum datagram (data section) length

Range of data : 80 to 256 (characters)
If a value out of the valid range is specified, either 80 or 256 is assumed.
The default value is 256 (characters).
This parameter defines a maximum packet length for DNC2 transmission. The maximum length of a packet is the length specified by parameter 5055 plus nine characters including the first two characters, four command characters, and the last three characters.



	#7	#6	#5	#4	#3	#2	#1	#0
5056				PCDAT	ERCODE	NCKER		NCKBCC

NCKBCC : Whether to check the BCC value
0 : Check
1 : No check; the BCC cannot be omitted.

NCKER : Whether to check the ER (RS-232-C)/TR (RS-422) signal
0 : Check
1 : No check

ERCODE : Error code for negative response
0 : Negative response is not indicated by an error code.
1 : Negative response is indicated by an error code consisting of four hexadecimal digits.

When using the FANUC-created DNC2 communication program library in the host computer, set this parameter to 1.

PCDAT : For PMC–NA, PMC data is read and written with:
0 : Address specification type.
1 : Data number specification type.

For PMC–NB/NB2, the data number specification type is always used regardless of the setting of this parameter.

B.3 FANUC Series 16/18/21

The following parameters are transferred to the DNC2 system only when the CNC is turned on. Therefore, when these parameter values have been changed, the power to the CNC must be turned off, then back on.

0020	I/O channel : I/O unit selection

When DNC2 is to be used, set 10 in this parameter.

	#7	#6	#5	#4	#3	#2	#1	#0
0131					ASI			SB2

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

ASI : Code used for data input is:
0 : ISO code.
1 : ASCII code.

NOTE
Use ASCII code.

	#7	#6	#5	#4	#3	#2	#1	#0
0134			CLK	NCD			PRY	

PRY : The parity bit is:
0 : Not used.
1 : Used (even parity).

NCD : The CD (signal quality detection) signal (RS–232C only) is:
0 : Checked.
1 : Not checked.

CLK : The baud rate clock (RS–422 only) is:
0 : The internal clock.
1 : The external clock.

NOTE
The data bit length is determined depending on the RSASC and PARTY settings. If RSASC and PARTY are both set to 1, the 7–bit length is used. In other cases, the 8–bit length is used.

	#7	#6	#5	#4	#3	#2	#1	#0
0135					R42			

R42 : As the interface:
0 : The RS–232C interface is used.
1 : The RS–422 interface is used.

	#7	#6	#5	#4	#3	#2	#1	#0
0140					ECD	NCE		BCC

BCC : The BCC value is:
0 : Checked.
1 : Not checked. (BCC itself cannot be omitted.)

NCE : The ER (RS–232C)/TR (RS–422) signal is:
0 : Checked.
1 : Not checked.

ECD : Error code for negative acknowledgment
0 : No error code is added to negative acknowledgment.
1 : A 4–digit hexadecimal error code is added to negative acknowledgment.

When using the DNC2 communication library, developed specifically by FANUC for the host computer, this parameter must always be set to 1.

0133	Baud rate

Valid data range : 1 to 12 (RS–232C), 5 to 15 (RS–422)
If a value that falls outside the valid data range is specified, the maximum or minimum value is set. The default is 10. (4800 bps).

Value	Baud rate	Value	Baud rate	Value	Baud rate
1	50	6	300	11	9600
2	100	7	600	12	19200
3	110	8	1200	13	38400
4	150	9	2400	14	76800
5	200	10	4800	15	86400

0143	Timeout value of the timer used to check for a response from the host

Valid data range : 1 to 60 seconds
If a value that falls outside the valid range is specified, 1 or 60 is assumed. The default is 5 seconds.

0144	Timeout value of the timer used to check for the EOT signal from the host

Valid data range : 1 to 60 seconds
If a value that falls outside the valid range is specified, 1 or 60 is assumed. The default is 5 seconds.

0146	Maximum allowable number of times a demand is made in the case of invalid protocol or no–response state in the data link layer

Valid data range : 1 to 10
If a value that falls outside the valid range is specified, 1 or 10 is assumed. The default is 5 times.

0147	Maximum allowable number of times a message is resent in response to NAK

Valid data range : 1 to 10
If a value that falls outside the valid range is specified, 1 or 10 is assumed. The default is 3 times.

0148	Maximum number of characters that can be received after send stop processing (CS off)

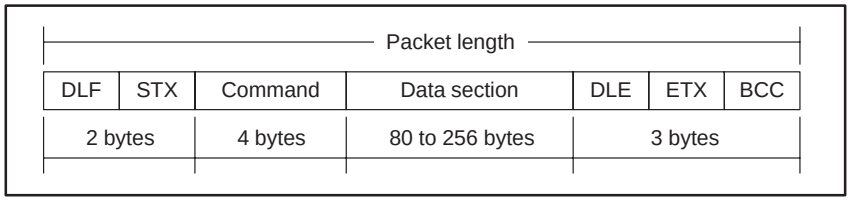
Valid data range : 10 to 255 characters
If a value that falls outside the valid range is specified, 10 or 255 is assumed. The default is 256 characters.

0149	Maximum length of datagram (data section)

Valid data range : 80 to 256 characters
If a value that falls outside the valid range is specified, 80 or 255 is assumed. The default is 256 characters.

In DNC2 sending, the maximum packet length is defined by this parameter.

The maximum packet length is the setting in [No. 0149] plus nine characters (the sum of the first two characters, the four characters in the command section, and the last three characters).



C DATA LINK LAYER MATRIX

Matrix state \ Event		Trns req.	Trns end	Received ENQ	Received DLE	Received "0"	Received "1"	Received NAK	Received ELSE	Time Out
		A	B	C	D	E	F	G	H	I
IDLE state	0	send ENQ 2	–	send DLE0 Na, Nb, Nc=0 7	ignore data 0	← same to D	← same to D	← same to D	← same to D	–
wait ENQ state (Host only)	1	–	–	send DLE0 Na, Nb, Nc=0 7	retry over Nb 1 / 0	← same to D	← same to D	← same to D	← same to D	0
TRANSMITTING: wait DLE state (DLE0)	2	– –	–	CNC send ENQ / HOST to wait ENQ / retry over Na 2 / 1 / 0	to wait "0" 3	send ENQ / retry over Na 2 / 0	← same to E	to IDLE state 0	← same to E	← same to E
TRANSMITTING: wait F0G state	3	–	–	CNC send ENQ / HOST to wait ENQ / retry over Nb 2 / 1 / 0	send ENQ / retry over Nb 2 / 0	start transmitt Na, Nb, Nc=0 4	send EOT ENQ / retry over Nb 2 / 1	← same to D	← same to D	← same to D
TRANSMITTING: transmitting state	4	–	to DLE wait 5	evaluate after transmitting is completed 4						–
TRANSMITTING: wait DLE state (DLE1)	5	–	–	send ENQ / retry over Na 5 / 0	to wait "1" 6	← same to C	← same to C	retry over Nc 4 / 0	← same to C	← same to C
TRANSMITTING: wait "1" state	6	–	–	send ENQ / retry over Na 5 / 0	← same to C	retry over Nc 4 / 0	send EOT (normal) 0	← same to E	← same to C	← same to C Ta

A	B
n	m

A, B : Process
m, n : Matrix state number to be changed

Na : Retry counter A
Nb : Retry counter B
Nc : NAK counter

Ta : No response timer
Tb : EOT timer

Matrix state \ Event		Trns req.	Trns end	Received ENQ	Recv. DLE	Recv. STX	Recv. ETX	Recv. BCC	Recv. EOT	Recv. ELSE	Time Out	
		A	B	C	D	E	F	G	H	I	J	
RECEIVING: wait DLE state (DLE STX)	7	–	–	send DLE0 / retry over Na 7 / 0	to STX wait 8	← same to C	← same to C	–	← same to C	← same to C	← same to C Ta	
RECEIVING: wait STX state	8	–	–	send DLE0 / retry over Nb 7 / 0	← same to C	to DLE wait 9	← same to C	–	← same to C	← same to C	← same to C Ta	
RECEIVING: wait DLE state (DLE ETX)	9	–	–	Regard as text 9	to ETX wait 10	← same to C	← same to C	–	← same to C	← same to C	send DLE0 / retry over Ta, Nb 7 / 0	
RECEIVING: wait ETX state	10	–	–	Regard as text 9	← same to C	← same to C	to BCC wait 11	–	← same to C	← same to C	send DLE0 / retry over Ta, Nb 7 / 0	
RECEIVING: wait BCC state	11	–	–	BCC valid : send DLE1 to EOT wait 12			BCC invalid : 7 send NAK 7		retry over Nc 0		send DLE0 / retry over Ta, Nc 7 / 0	
RECEIVING: wait EOT state	12	–		send DLE1 / retry over 12 / 0	← same to C	← same to C	← same to C	–	to IDLE 0	← same to C	to IDLE Tb 0	

D APPLICATION LAYER MATRIX

1. PT**/LIPM type

Event / Matrix state		Receive req.	Received PT**/LIPM		Received T NB		Recv. R **/ DIPM	Recv. M OK	Recv. M NR	Recv. T NP	Recv. T BD	Recv. M RR	Recv. M RT	Recv. T FD	Recv. else	CNC reset	Datalink error
		A	B		C		D	E	F	G	H	I	J	K	L	M	N
IDLE state	0	PT ** send	M RT send	M NR/ T NP send	T BD send						No opera-tion	T BD send				No operation	
		3	1	0	0						0	0				0	
T NB wait	1	–	T BD send		R **/ DIPM send	T FD send	← same to B				to IDLE	← same to B				T BD send	to IDLE
			0		1	2					0					0	0
M OK wait	2	–	T BD send		← same to B			to IDLE	← same to B		to IDLE	← same to B				T BD send	to IDLE
			0					0			0					0	0
M RT wait	3	–	T BD send		← same to B				to IDLE	to IDLE	to IDLE	← same to B	T NB send	← same to B		T BD send	to IDLE
			0						0	0	0		4			0	0
R ** wait	4	–	T BD send		← same to B		T NB send	← same to B		to IDLE	to IDLE	← same to B		M OK send	← same to B	T BD send	to IDLE
			0				4			0	0			0		0	0

A	B
n	m

A, B : Process
m, n : Matrix state number to be changed

2. PR** type

Event / Matrix state		Send req.	PT**/LIPM		T NB		R **	M OK	M NR	T NP	T BD	M RR	M RT	T FD	else	CNC reset	Datalink error
		A	B		C		D	E	F	G	H	I	J	K	0	M	N
IDLE state	0	PR ** send	M PR send	M NR M NP send	T BD send						No opera-tion	T BD send				No operation	
		2	1	0	0						0	0				0	
R ** wait	1	–	T BD send		← same to B		T NB send	← same to B		to IDLE	to IDLE	← same to B		M OK send	← same to B	T BD send	to IDLE
			0				1			0	0			0		0	0
M RR wait	2	–	T BD send		← same to B				to IDLE	← same to B	to IDLE	PR ** send	← same to B			T BD send	to IDLE
			0						0		0	3				0	0
T NB wait	3	–	T BD send		R ** send	T FD send	← same to B				to IDLE	← same to B				T BD send	to IDLE
			0		3	4					0					0	0
M OK wait	4	–	T BD send		← same to B			to IDLE	← same to B		to IDLE	← same to B				T BD send	to IDLE
			0					0			0					0	0

3. T** type

Event / Matrix state		Send Req.	T **		R **		M OK	M NR	T NP	T BD	else	CNC reset	Datalink error
		A	B		C		D	E	F	G	H	I	J
IDLE state	0	T ** send	R ** send	M NR/ T NP send	T BD send					No operation	T BD send	No operation	
		2	1	0	0					0	0	0	
M OK wait	1	–	T BD send		← same to B		to IDLE	to IDLE	← same to B	to IDLE	← same to B	T BD send	to IDLE
			0				0	0		0		0	0
R ** wait	2	–	T BD send		M OK send	M NR send	← same to B			to IDLE	← same to B	T BD send	to IDLE
			0		0	0				0		0	0

4. M**/R** type

Event / Matrix state		Send Req.	M ** /MCRM R **		M NR	M NP	T BD	else	CNC reset	Datalink error
		A	B		C	D	E	F	G	H
IDLE state	0	R ** / M ** send	M OK send	M NR M NP send	T BD send	T BD send	No operation	T BD send	No operation	
		1	0	0	0	0	0	0	0	
M OK wait	1	–	T BD send		normal to IDLE	normal to IDLE	to IDLE	← same to B	T BD send	to IDLE
			0		0	0	0		0	0

E ALARM MESSAGES

E.1 Series 0

Table E.1 Alarm Messages for Series 0

Code	Message	Description	Remark
BP/S 071	BP/S 071 alarm	A program specified to be deleted or uploaded was not found.	
BP/S 073	BP/S 073 alarm	There is a program with the same name as a program specified to be downloaded.	
BP/S 086	BP/S 086 alarm	Errors that disable communication (1) Parity error (2) Flaming error (3) Overrun error (4) CD (signal quality detection) low (5) DR/TR signal low	Note

NOTE
Recovery from this error can be done by turning off and on again the ER/TR signal in the host computer or turning off the power to the CNC.

E.2 Series 15

Table E.2 Alarm Messages for Series 15

Code	Message	Description	Remark
SR 960	MAP file transmission error (channel 1)	A communication error occurred on channel 1. a) The operator or the host computer discontinued transmission. b) Errors that disable communication (1) Parity error (2) Flaming error (3) Overrun error (4) CD (signal quality detection) low (5) DR/TR signal low	Note
SR 970	MAP file transmission error (channel 2)	Same as above	

NOTE
Recovery from this error can be done by turning off and on again the ER/TR signal in the host computer or turning off the power to the CNC.

F ERROR CODES DETAILING NEGATIVE RESPONSE

Fig.F Error Codes Detailing Negative Response (1/2)

Error code	Meaning	Remark
FEBAH–046H	Command syntax error	T_ER
FFB9H–047H	Command exchange sequence error	M_BD
FFCEH–032H	Negative program number other than =9999	M_NR
FFCDH–033H	Program number format error	N_NR
FFCCH–034H	Specified axis has not been mounted.	M_NR
FFCBH–035H	Too many data items were requested.	M_NR
FFCAH–036H	No data item can be read.	M_NR
FFC9H–037H	Invalid axis command	M_NR
FFC6H–03AH	A request is being processed.	T_BD
FFC5H–03BH	A request is being canceled.	T_BD
FF00H–100H	The CNC makes an interrupt request.	T_BD
FDFFH–201H	Timeout	M_NP
FDFEH–202H	Channel busy	M_NP
FDFDH–203H	Data remaining	M_NP
FDFCH–204H	Incorrect file name	T_BD
FDFBH–205H	Open request rejected	M_NP
FDFAH–206H	Edit request rejected	M_NP
FDF9H–207H	CNC busy	M_NP
FC0CH–3F4H	A specified file was not found.	M_NR
FC0BH–3F5H	A warning occurred during selection of a file.	T_NP
FC0AH–3F6H	Start request rejected	M_NP
FC09H–3F7H	Not in auto mode	M_NP
FC08H–3F8H	File selection request rejected	M_NP
FC07H–3F9H	File deletion request rejected	M_NP
FC06H–3FAH	File protected	M_NP
FC05H–3FBH	File deletion request rejected, or warning occurred during file deletion.	M_NP
FC04H–3FCH	Editing request rejected	M_NP
FC03H–3FDH	Directory not found	T_NP
FC02H–3FEH	Directory read request rejected	T_NP
FC01H–3FFH	Invalid directory or file name	T_BD
FBA8H–458H	Window library error	M_NP
FBA7H–459H	Command not executable	M_NP
FBA6H–45AH	Invalid function code	T_BD
FBA5H–45BH	Invalid major data classification	M_NR
FBA4H–45CH	Invalid medium data classification	M_NR
FBA3H–45DH	Invalid minor data classification	M_NR
FBA2H–45EH	Invalid data length	T_BD
FBA1H–45FH	Invalid data type	T_BD
FBA0H–460H	Invalid data	M_NP
FB9FH–461H	Miscellaneous command errors	M_NP
FB9EH–462H	No option	M_NP

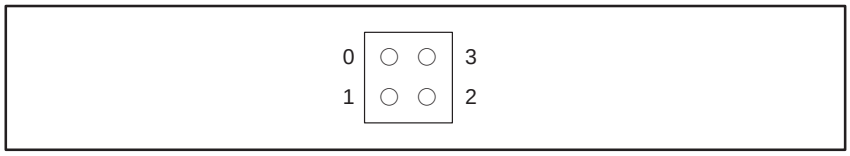
Fig.F Error Codes Detailing Negative Response (2/2)

Error code	Meaning	Remark
FB9DH–463H	File not found	M_NR
FB9CH–464H	File protected	M_NP
FB9BH–465H	No directory space	M_NP
FB9AH–466H	Insufficient memory space	M_NP
FB99H–467H	Read disabled	T_NP
FB98H–468H	Write disabled	M_NP
FB97H–469H	Write failed	T_BD
FB96H–46AH	Read failed	T_BD
FB95H–46BH	Device inoperable	M_NP
FB94H–46CH	System not ready	M_NP
FB93H–46DH	Invalid status	M_NP
FB92H–46EH	Data mismatch	T_BD
FB46H–4BAH	No extended window option	M_NP
FB45H–4BBH	Miscellaneous errors	M_IL
F62DH–9D3H	Write protected	M_NP
F62CH–9D4H	Protect key locked	M_NP
F62BH–9D5H	Invalid mode	M_NP
F62AH–9D6H	Invalid address	M_NP
F629H–9D7H	Data outside valid range	M_NP
F628H–9D8H	Too many digits	M_NP
F627H–9D9H	Start disabled	M_NP
F626H–9DAH	Input of this parameter is inhibited.	M_NP
F625H–9DBH	Data not found	M_NR
F624H–9DCH	Background editing in progress	M_NP
F623H–9DDH	External I/O unit channel being used	M_NP
F622H–9DEH	No program selected	M_NP
F621H–9DFH	Editing impossible	M_NP
F620H–9E0H	System error	M_NP
F61FH–9E1H	A program already exists.	M_NR
F61EH–9E2H	Insufficient memory area	T_BD
F61DH–9E3H	Not in emergency stop state	M_NP
F61CH–9E4H	Already being used	M_NP
F61BH–9E5H	Too many programs	M_NP
F61AH–9E6H	Outside valid range	M_NP
F619H–9E7H	Editing of this word not allowed	M_NP
F618H–9E8H	No program number	M_NP
F617H–9E9H	Command rejected	M_NP
F616H–9EAH	Running program	M_NP
F615H–9EBH	Program not allowed to be displayed	M_NP
F614H–9ECH	No option found	M_NP
F60BH–9F5H	Data missing	M_NP
F606H–9FAH	Background editing rejected	M_NP
F605H–9FBH	Parameter setting error	M_NP
F604H–9FCH	Background NC status error	M_NP

G MAINTENANCE

G.1 DESCRIPTIONS OF INDICATION BY LEDS (Series 15–A)

The arrangement of the LEDs is as shown on the right (the door open). The following table lists the meanings of the LED indications.



G.1.1 Blinking LED Indications

When the LEDs are blinking or remain off, they indicate system errors. The system errors can normally be eliminated by turning the power off and on again. In some cases, error recovery requires hardware replacement.

Table G.1.1 Errors Indicated by Blinking LEDs

No.	LED indication	Meaning	Code	Message
1	○ ○ ○ ○	An NMI occurred in a module other than the remote buffer.	F	No
2	● ○ ○ ○	RAM parity error	E	Yes
3	○ ○ ● ○	F-BUS error	D	Yes
4	● ○ ● ○	ROM parity error	C	No
5	○ ○ ○ ●	Invalid interrupt An interrupt occurred when should not.	B	Yes
6	● ○ ○ ●	Error on the ID of a remote buffer PCB	A	No
7	○ ○ ● ●	RAM collation error	9	No
8	● ○ ● ●	Not used	8	—
9	○ ● ○ ○	DNC2 software error	7	No

● ... Off
○ ... On

If the LEDs are on or off with one of the patterns listed above, check the information described below using the CNC diagnostic menu and report it to the maintenance personnel: Four words at addresses 46003FEF to 46003FEB (selected from among addresses 46003F80H [46003F00H] and on) displayed on the screen using INP-NO; enclosed in brackets are the addresses that can be displayed on a 14-inch CRT screen.

G.1.2 Nonblinking LED Indications

When the LEDs light steadily or remain off, they indicate the state (not abnormal state) of the remote buffer.

Table G.1.2 LED Indications and Their Meanings

No.	LED indication	Meaning	Code	Message
1	○ ○ ○ ○	The remote buffer CPU is at halt, immediately after power is applied.	F	
2	○ ● ● ●	CIRR#15=1 wait; Common Resource Irt. request awaited	1	
3	● ● ○ ●	CIRR#15=0 wait after the remote buffer SFT has been initialized	2	
4	○ ● ○ ●	CNC's PWN#2=1 wait after the remote buffer module has been initialized; for all modules, completion of MDL setting awaited.	3	
5	● ● ● ○	CIRR#15=1 wait; completion of IPL awaited	4	
6	○ ● ● ○	CIRR#15=0 wait after remote buffer's PWN#3 = 1; for all modules, completion of MDL setting awaited	5	
7	● ● ○ ○	Wait for a request (command) from the CNC	6	
8	○ ● ○ ○	DNC2 software is ready.	7	

G.2 SYSTEM ERROR MESSAGES (Series 15–A)

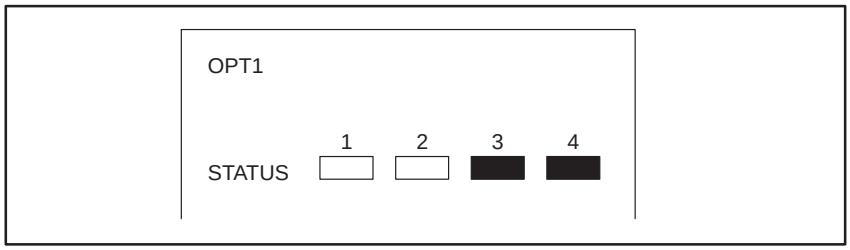
Incase of Series 15–A, the following table lists the system error messages related to the remote buffer.

Table G.2 LED Indications and Their Meanings

LED indication	Code	Meaning	Message
● ○ ○ ○	E	RAM PARITY LH (000aaaaa)	RAM parity error occurred. LH ... L : Even address H : Odd address aaaaa Address (next command) detected when the error occurred
○ ○ ● ○	D	BUS ERROR (000aaaaa)	F-BUS error occurred. aaaaa Address (next command) detected when the error occurred
○ ○ ○ ●	B	UNDEFINED IRT 00n (000aaaaa)	Invalid interrupt occurred. 000n... Type of the interrupt aaaaa Address (next command) detected when the error occurred

G.3 LED INDICATORS (Series 15–B)

The Series 15–B has LEDs arranged as shown in the figure on the right (as viewed with the door open). LEDs 3 and 4 are used for DNC2. Immediately after the power is turned on, LEDs 3 and 4 both light while the DNC2 CPU is stopped; then, these LEDs normally go off.



The following LED states indicate system errors. Such errors cannot be recovered unless the power is turned off then back on. Some errors require hardware replacement.

No.	LED status No.3	LED status No.4	Meaning	Internal code	Message
1	On	Blink	NMI is generated in a module other than DNC2.	8F	Not provided
2	Blink	On	RAM parity error	8E	Provided
3	Blink	Blink	F–bus error	8D	Provided
4	Blink	Off	ROM parity error	8C	Provided
5	Off	Blink	RAM check error	89	Provided
6	Blink / Off (LEDs 3 and 4 blink alternately.)	Off / Blink	This state may indicate the following problems. To determine the problem, see the message displayed on the CRT screen. a) DNC2 printed circuit board failure <<Applicable to version B and later printed circuit boards>> b) Invalid interrupt (An illegal interrupt is generated.) c) DNC2 software error	87 88 8A 8B	Provided

For details of the messages, see Appendix G.4.

If any of the above LED states is observed, record the four words (46800004 to 4680000B) on from the CNC diagnostic screen, and contact FANUC.

NOTE
DNC2 RAM ranges from 800000H to 8FFFFFH. Do not try to display 46900000 and subsequent memory locations. Otherwise, a system error occurs.

G.4 SYSTEM ERROR MESSAGES (Series 15–B)

With the Series 15–B, the system error messages related to DNC2 are explained below.

Internal code	Message	Meaning
8E	RAM PARITY HL (00aaaaaa)	RAM parity error occurred. LH... L : Even–numbered address H : Odd–numbered address aaaaaa Execution address when the error occurred (next instruction)
8D	BUS ERROR (00aaaaaa)	F–bus error occurred. aaaaaa... Execution address when the error occurred (next instruction)
8C	DNC2 ROM PARITY ERROR	ROM parity error occurred.
8B	UNDEF IRT 00nn (00aaaaaa) ERR–CODE : cccc	Invalid interrupt was generated. 00nn... Interrupt type aaaaaa... Execution address when the error occurred cccc... Error code
8A	DNC2 PCB HARD ERROR (diimage)	Problem occurred in DNC2 printed circuit board version B or later. diimage...Status of 7000n4h in communication space
89	DNC2 RAM TEST ERROR (seg : addr)	RAM check error occurred. seg... Error segment addr... Error address
88 87	DNC2 SOFT ERROR (id : ee : aaaa)	DNC2 software error occurred. id... Error type ee... Error internal code aaaa... Execution address when the error occurred (next instruction)

G.5 TROUBLESHOOTING

G.5.1 MAP Log Messages

G.5.1.1 Log messages

Datagram communication is always recorded in the log buffer area. The most recent 500 characters recorded in the log area are displayed on the CNC MAP log message screen. Figure G.5.1.1 shows a log message display. This function is supported only by the Series 15.

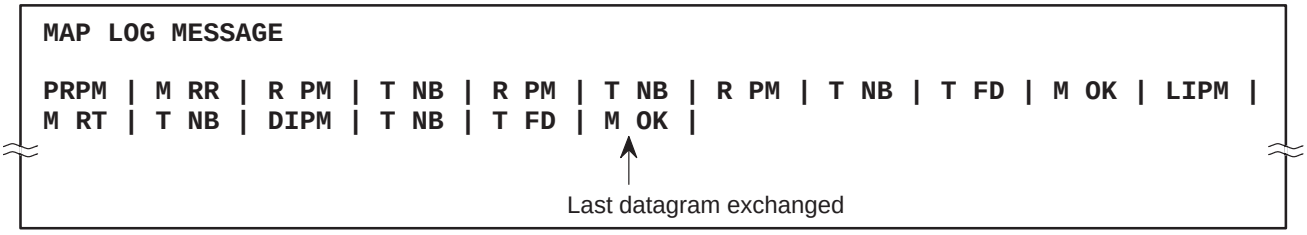


Fig.G.5.1.1 Log Display

G.5.1.2 Log message format

There are three log message formats. Each log message is separated by ASCII character " | ". The meaning of the messages in each format is as follows:

1) **<CCCC>** :
   CCCC Command section of datagram
2) **<NNNN0Xeeee>**:
   NNNN Negative response (datagram command section)
   eeee Error code (See Appendix F.)
3) **<Ennn0Xssdd>**:
   Ennn Data link layer error code
   See Table G.3.1.2.
   ssdd Details of the error
   ss : Status of the data link layer matrix
   dd : Error data

Table G.5.1.2 Error codes

Error code	Details (0xssdd)		Explanation
	Status (ss)	Error data (dd)	
E002	0, 1, 2, 3 5, 6, 7, 8 9, A, B, C	Error code of communication error	Communication line disconnection
E009	2, 3, 5, 6 7, 8, 9, A B	00	Retry limit reached due to timeout
E00C	0, 1, 2, 3 5, 6, 7, 8 C	Invalid TCC(*1)	Retry limit reached due to reception of invalid character
E010	5, 6, B	Error code 5 : "NAK" 6 : "NAK" | "DLFO" B : "BCC"	Retry limit reached due to NAK or DLE0 Retry limit reached due to checksum error
E011	B	00	Wait for BCC
E014	C	Invalid TCC(*1)	Retry limit reached due to reception of invalid character during wait for EOT

*1 TCC stands for Transmission Control Character.

G.5.2 Troubleshooting

No.	Symptom	Check procedure
1.	Commands cannot start operation. A prerequisite for normal operation is that the following parameters have been set correctly. (1) Baud rate (2) Stop bits (1 or 2) (3) External clock baud rate (4) Parity (no parity or even parity) (5) Transmission code (ASCII/ISO) (6) Interface (RS-232-C/RS-422) (7) Maximum datagram length (8) Interface number Series 0 : I/O=10 Series 15: No. 20, 21, 22, 23=4 **(Note)** If a parameter value is out of the valid range, the default value is used. Check the default value for each device.	Troubleshoot according to the following procedure. a) Check whether the LED indication pattern is No. 8 described in G.1.2. b) Check the MAP log message. If no log message is displayed, check whether: (1) Parameters have been set. (2) The power was turned off and on again after the parameters were changed. (3) The communication is in a satisfactory condition. If a log message is displayed, perform the following: (1) Eliminate the cause of error Ennnn0Xssdd (if displayed). (2) If a command exchange sequence has not been completed, check which device is the requester or the responder.
2.	Operation was discontinued halfway through.	Perform the same check as No.1 except for checking of parameters.
3.	Alarm SR960/070 (Series 15) or BP/S86 (Series 0/16/18/21) occurred.	Check the following: a) Communication cable and connectors b) Flaming or overrun error (1) Baud rate mismatch (2) Improper stop bit setting (3) CD (signal quality detection) is off (RS-232-C only).

H MODAL DATA AND CORRESPONDING INTERNAL CODES

Group (gg)	G code		Internal code (vv)
	Machining center	Lathe (system B)	
1	G00	G00	0
	G01	G01	1
	G02	G02	2
	G03	G03	3
	G33	G33	4
	——	G77	8
	——	G78	9
	——	G79	10
2	G17	G97	0
	G18	G96	1
	G19	——	2
3	G90	G90	0
	G91	G91	1
4	G22	G22	0
	G23	G23	1
5	G93	——	2
	G94	G94	0
	G95	G95	1
6	G20	G20	1
	G21	G21	0
7	G40	G40	0
	G41	G41	1
	G42	G42	2
8	G43	——	1
	G44	——	2
	G49	——	0
9	G73	G83.1	10
	G74	G84.1	11
	G76	G86.1	12
	G80	G80	0
	G81	G81	1
	G82	G82	2
	G83	G83	3
	G84	G84	4
	G85	G85	5
	G86	G86	6
	G87	G87	7
	G88	G88	8
	G89	G89	9

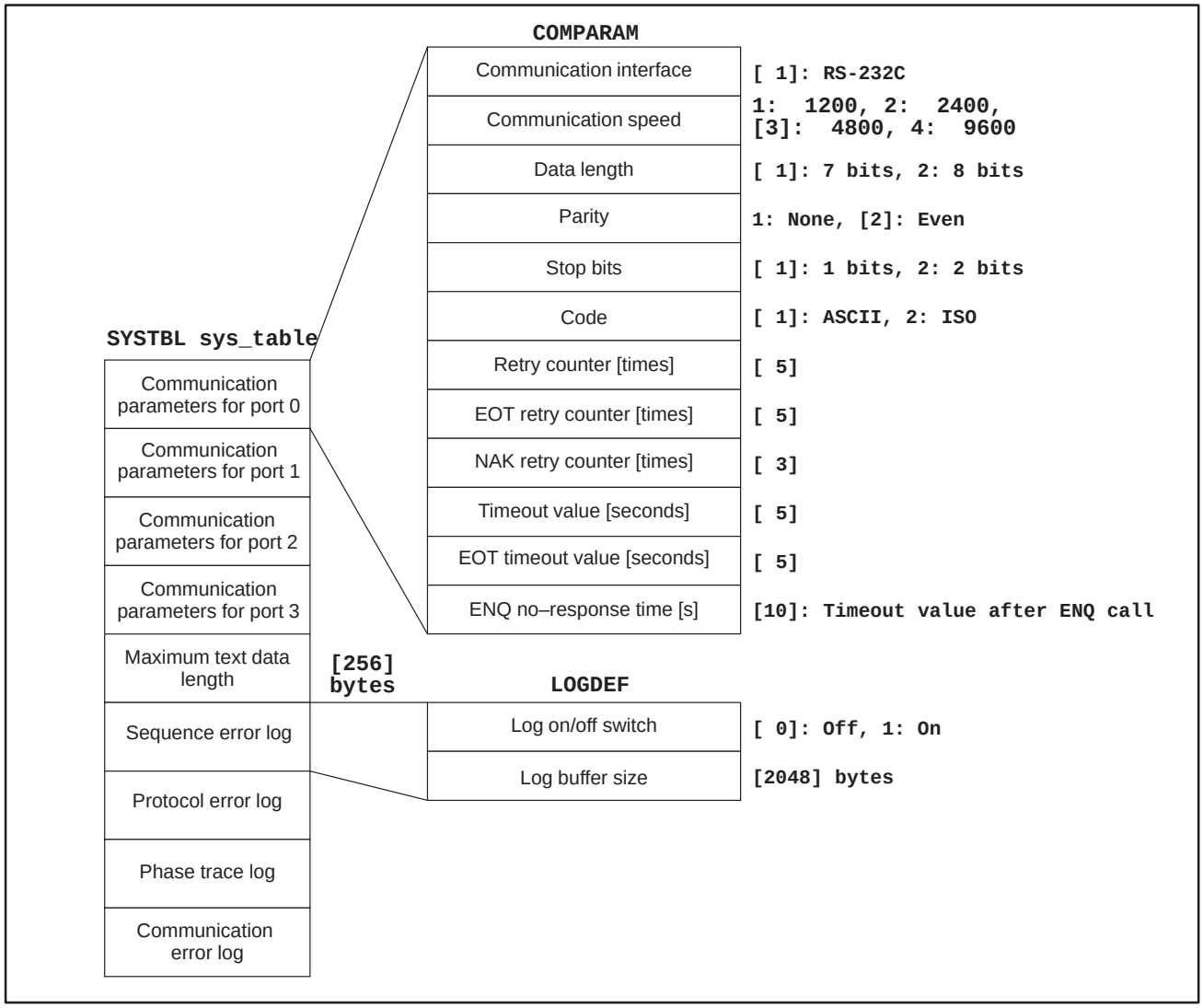
Group (gg)	G code		Internal code (vv)
	Machining center	Lathe (system B)	
10	G98	G98	1
	G99	G99	0
11	G50	——	0
	G51	——	1
12	G66	G66	1
	G66.1	G66.1	2
	G67	G67	0
13	G96	G68	1
	G97	G69	0
14	G54	G54	0
	G55	G55	1
	G56	G56	2
	G57	G57	3
	G58	G58	4
	G59	G59	5
15	G61	G61	1
	G62	G62	2
	G63	——	3
	G64	G64	0
16	G68	G18	1
	G69	G17	0
	——	G19	2
17	G15	——	0
	G16	——	1
18	G50.1	G50.1	0
	G51.1	G51.1	1

Note) With lathe machines, the same G code is output for the corresponding G codes of individual systems.

COMMUNICATION PARAMETER STRUCTURE

The following diagram shows the initial values and other valid values for the communication parameters such as communication speed, parity, and timeout in the system parameter table <**sys_table**>. The initial values are enclosed in braces.

The parameter settings described here must coincide with those in the CNC.



NOTE
In the communication library provided by FANUC, the maximum transmission speed is 9600 bps.

```
typedef  struct    {
    COMPARAM  port[4];
    short    len;
    LOGDEF   seqlog;
    LOGDEF   prolog;
    LOGDEF   phalog;
    LOGDEF   comlog;
}  SYSTBL;
```

```
typedef  struct    {
    char   type;
    char   speed;
    char   len;
    char   parity;
    char   stop;
    char   code;
    short  retry;
    short  eotrty;
    short  nakrty;
    short  tmout;
    short  eotout;
    short  enqout;
}  COMPARAM;
```

```
typedef  struct    {
  short  sw;
  short  size;
}  LOGDEF;
```

J ERROR CODES

(1) Error numbers 0FFFFh to 0FFCFh (absolute values 1 to 49)

Code	Name	Description
0FFFFh	Communication error	Error related to communication.
0FFFEh	File open error	A specified file was not found.
0FFFDh	File read error	A specified file cannot be read from, for example, because the disk is damaged.
0FFFCh	File write error	A specified file cannot be written to, for example, because there is no free space.
0FFF0h	Sequence error	A received command has not been defined in the protocol.
0FFFBh	Error upon reception of negative acknowledgment Abnormal number	When M_NR was received from the CNC, there was no error number, or the error number was abnormal.
0FFFAh		When M_NP was received from the CNC, there was no error number, or the error number was abnormal.
0FFF9h		When T_NP was received from the CNC, there was no error number, or the error number was abnormal.
0FFF1h		When T_BD was received from the CNC, there was no error number, or the error number was abnormal.
0FFEFh		When T_FD was received from the CNC, there was no error number, or the error number was abnormal.
0FFEEh		When M_ER was received from the CNC, there was no error number, or the error number was abnormal.
0FFEDh		When M_IL was received from the CNC, there was no error number, or the error number was abnormal.
0FFEB		Syntax error of datagram received from the CNC

(2) Error numbers 0FFCFh and on (absolute values 50 and on)
See descriptions on error codes for negative response in Appendix F.

K DEMONSTRATION PROGRAM

K.1 OUTLINE OF FUNCTIONS

This demonstration program is supplied as a sample program of the communication library. This program uses several communication library functions (the downloading, program deletion, and PMC data read/write functions) to perform simple schedule operation in memory mode.

Since the program requires the Series 0–MC and a special ladder, merely executing the program on the supplied floppy disk does not allow functions 4, 5, 6, 8, and 9 in the command list to be performed correctly.

K.2 PREPARATION

K.2.1 Files

Copy the following files and the communication program library to the directory where you are going to compile the program.

No.	File	Outline of function
1	DEMO_MAN.C	Main routine of the demonstration program
2	D2_DEMO.C	Subfunctions of the demonstration program
3	D2_UTL98.C	Machine-dependent utility routines (for NEC PC-9801) in the demonstration program
4	D2_UTIBM.C	Machine-dependent utility routines (for IBN PC) in the demonstration program
5	D2DTGLB.C	Message tables for the demonstration program
6	D2DTEXT.H	Header file (external message table declarations) for the demonstration program
7	D2_DEMO.H	Header file (constant declarations) for the demonstration program
8	D2_GLB.H	Header file (function prototype declarations) for the demonstration program
9	D2_PRT.H	Header file (external function definitions) for the demonstration program
10	DNC2DEF.H	Header file (communication package functions and type declarations) for the demonstration program
11	DNC2SYS.CFG	File containing communication parameter settings

K.2.2 Compiling and Linking Programs

(1) Preparation

Set up environment variables while referring to setting of environment variables described in III–3.3.

(2) Compiling and linking

The communication program library contains makefiles for MAKE.EXE supplied together with the MS-C compiler. Compile your program using a makefile designed for your computer.

Makefile for IBM PC : IBM_DEMO.MAK
Makefile for NEC PC-98 : 98_DEMO.MAK
Makefile for Fujitsu FMR : FMR_DEMO.MAK

For the IBM PC, enter:

```
C>MAKE IBM_DEMO.MAK
```

An executable file will be generated automatically.
The executable file names are:

DEMOIBM.EXE for IBM PC
DEMO98.EXE for NEC PC-98
DEMOFMR.EXE for Fujitsu FMR

K.2.3 Setting the Communication Parameters

When the demonstration program is activated, it automatically sets the communication parameters as defined in DNC2SYS.CFG.
The user can change the communication parameter settings in DNC2SYS.CFG using a text editor. Any information other than parameter settings in DNC2SYS.CFG may not be changed, however. Be careful especially on the character strings, such as HspeedI, in the beginning. Changing them may hamper normal operation.

The details of the contents of the file are as follows:

(1) How to set the DNC2 system communication parameters

port=0 : Indicates that the following settings belong to port 0.
To generate the parameters for port 1, duplicate all information from Hport = ...I to HEOT timeout timeI.

type=1 : Communication interface (RS-232-C) cannot be changed.

speed=3 : Specifies communication speed by number.
1 : 1200bps
2 : 2400bps
3 : 4800bps
4 : 9600bps
Number 1 means a data signal speed of 1200bps.
Select a number from among those listed above that meets your data communication requirements.

char=1 : Specifies the length of a data character.
1 : 7 bits
2 : 8 bits
Number 1 means the data character is 7 bits long.
Select number 1 or 2 whichever meets your requirements.

parity=2 : Specifies parity by number.
1 : None
2 : Even
Number 1 means no parity bit is used.
Number 2 specifies even parity.
Select number 1 or 2 whichever meets your requirements.

stop bit=1 : Specifies the number of stop bits by number
1 : 1 bit
2 : 2 bits
Number 1 means one bit is used to indicate the end of a transmitted character.
Select number 1 or 2 whichever meets your requirements.

code=1 : Specifies a character coding system used by the NC equipment.
1 : ASCII
2 : ISO
Number 1 means ASCII.
Select number 1 or 2 whichever meets your requirements.

EOT retry count=5 : Specifies the EOT retry count by value.
A value of 0 means retry is repeated infinitely.

NAK retry count=3 : Specifies the NAK retry count by value.
A value of 0 means retry is repeated infinitely.

retry count=5 : Specifies the retry count by value.
A value of 0 means retry is repeated infinitely.

timeout time=5 : Specifies timeout interval (seconds) by value. A value of 0 means timeout will not occur.

EOT timeout=5 : Specifies the EOT timeout (seconds) by value. A value of 0 means timeout will not occur.

(2) How to set the maximum text data length for the data link layer

Text max lenth=256 : Specifies the maximum text data length for the data link layer by value (80 to 256).

(3) How to set up log conditions

To change log conditions, modify values set in the log setting section.

log sequence error : Sequence error log
See the descriptions below for how to set the parameters.

log protocol error : Protocol error log
See the descriptions below for how to set the parameters.

log phase trace : Phase trace log
See the descriptions below for how to set the parameters.

log communication error : Communication error log
See the descriptions below for how to set the parameters.

switch=0 : Specifies by number whether to turn the log switch on or off.
0 : Not logging
1 : Logging
Number 0 disables logging.
Number 1 enables logging.
This is an option for program debugging.
It should not be used for other purposes.

buff size=2048 : Specifies the maximum log file size in bytes.

K.3 OVERVIEW OF OPERATIONS

K.3.1 Starting and Exiting the Demonstration Program

After making sure that the executable file containing the demonstration program (DEMOIBM.EXE or DEMO98.EXE) and the communication parameter file (DNC2SYS.CFG) are on the current directory, activate the executable file.

Example) `C>DEMOIBM`

After a while, the menu appears and command entry is enabled.

To exit the demonstration program, select 10 : End on the command entry menu. If machining is in progress, press the ESC key to display the menu, and select 5 : Stop to stop machining. Then, select 10 : End.

K.3.2 Commands

No.	Command	Summary	Remark
1	Set new program	Registers a program.	Invalid during machining
2	Edit program	Makes changes regarding a registered program.	Invalid during machining
3	Delete program	Deletes a registered program.	Invalid during machining
4	Start	Starts machining.	
5	Stop	Stops machining.	
6	Continue	Resumes machining	Valid only during machining
7	Upload	Uploads a program.	Invalid during machining
8	Read PMC data	Reads PMC data.	
9	Write PMC data	Writes PMC data.	
10	End	Ends demonstration.	

K.3.3 Operating Procedure

(1) Immediately after the demonstration program is started, the command entry menu appears (see K.4.2). When machining is in progress, pressing the ESC key causes the program to wait for command entry.

(2) When the program is in command entry mode, register as many programs as necessary.

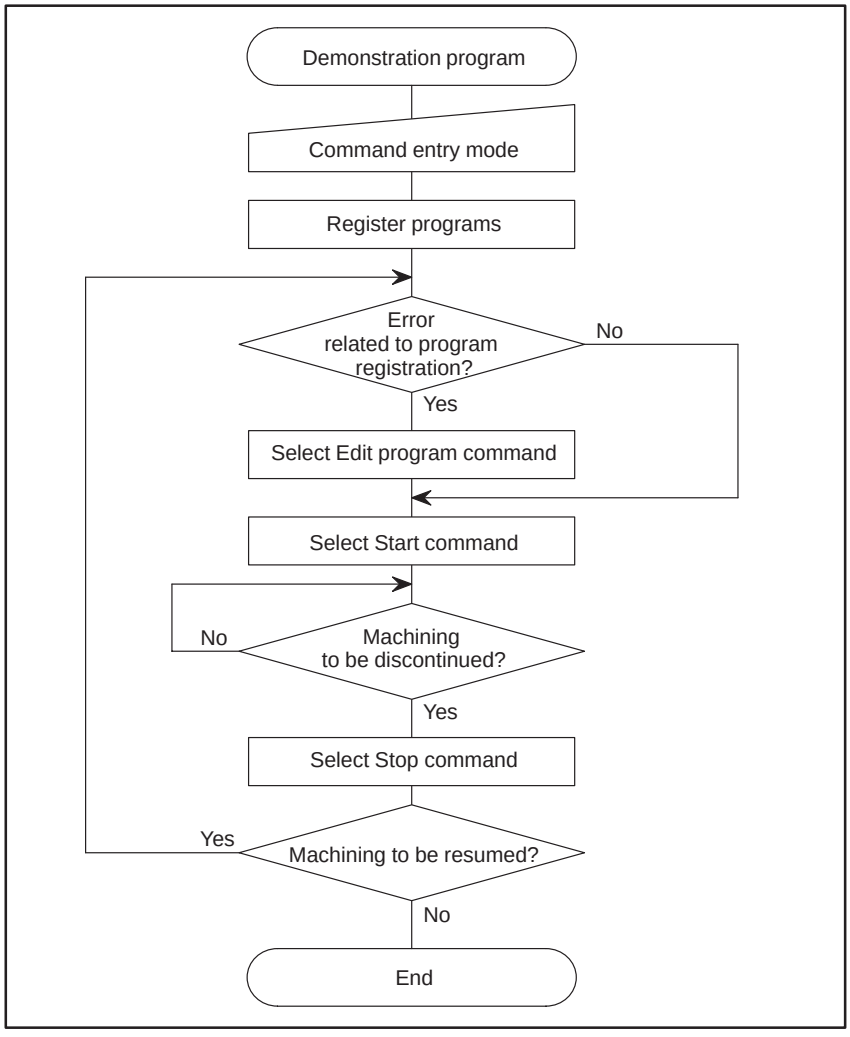
(3) If a program name or other information you entered is incorrect, correct it using the Edit program command. To delete a registered program, use the Delete program command.

(4) After all necessary programs have been registered, select the Start command to trigger machining. This causes the programs to be downloaded in sequence they were registered.

(5) When a program ends running (always monitored by the Read PMC data command), it is deleted from the NC memory, and the next program is downloaded to start machining.

(6) To discontinue machining, press the ESC key to place the program in command entry mode. Then select the Stop command.

(7) To read out the parameters during machining, press the ESC key to place the program in command entry mode. Then, select the Read PMC data command. To resume machining, select the Continue command.



K.3.4 Machining Procedure

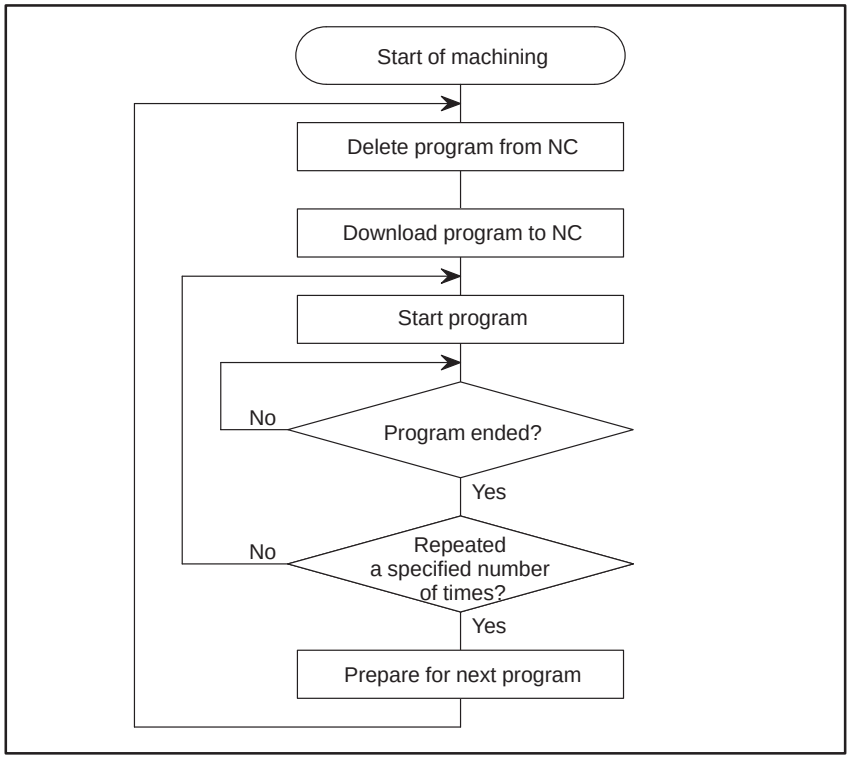
(1) Select the Start command. The lowest-numbered registered program is downloaded to the NCC.

(2) After the program is downloaded, start it and wait for it to end.

(3) When the end of program execution is detected, the program is restarted and executed as many times as specified (repetition count).

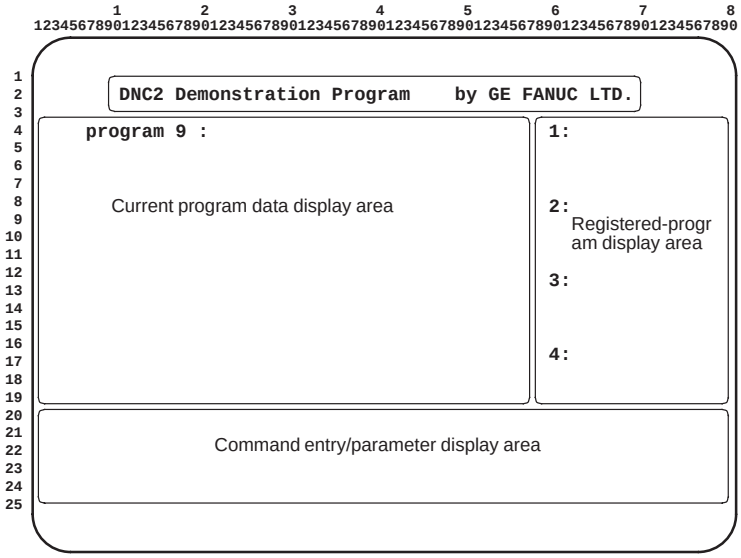
(4) After the program is executed a specified number of times, it is deleted from the NC, and the next program is downloaded.

(5) Steps 2 to 4 are repeated.



K.4
MENU LAYOUT

K.4.1
Main Menu Format



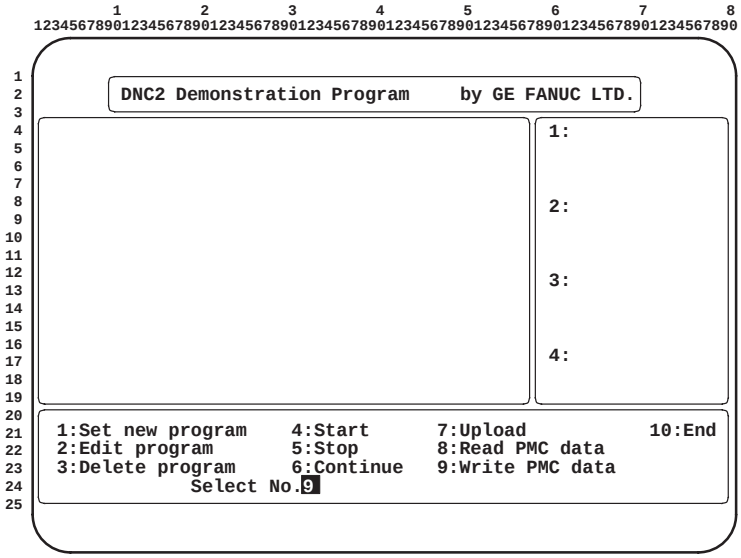
Registered-programs display area :
Displays the names of the registered programs; up to four programs can be registered.

Current program data display area :
Displays the details of the program being currently executed (how many times it has been executed and how long it took to execute it).

Command entry/parameter display area :
Displays the command and parameters entered.

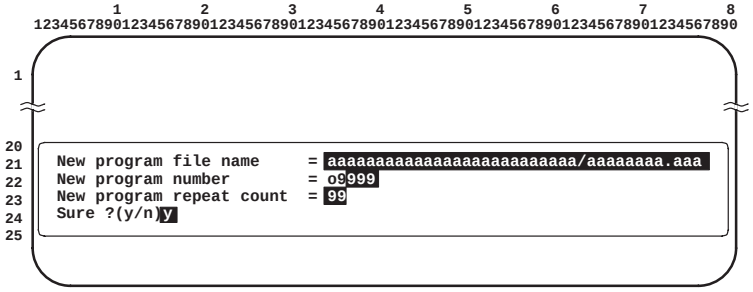
K.4.2 Command Entry

(1) Menu



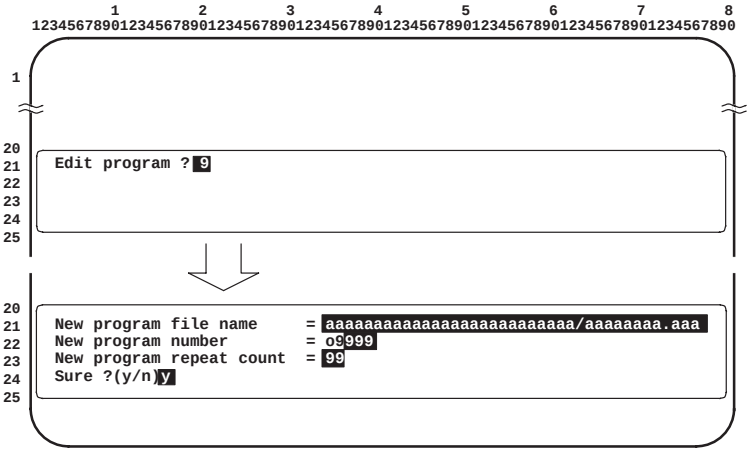
- This menu appears, when the demonstration program is started or when the ESC key is pressed during machining.
- Entering a number at the right of Select No. causes the next menu to appear.

(2) Set program command



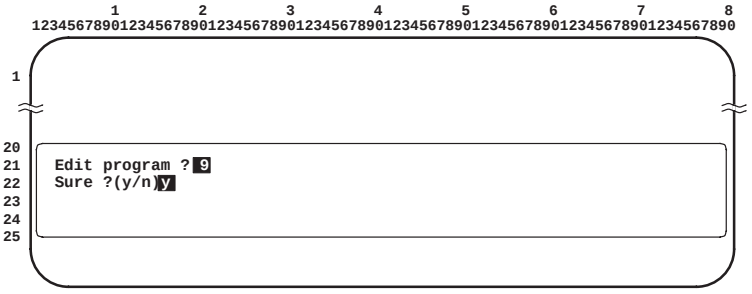
- Enter the name of a file containing a program to be registered, the number of the program, and the number of times (repetition count) that the program is to be executed.
- After a program has been registered, its name is displayed in the registered-program display area.

(3) Edit program command



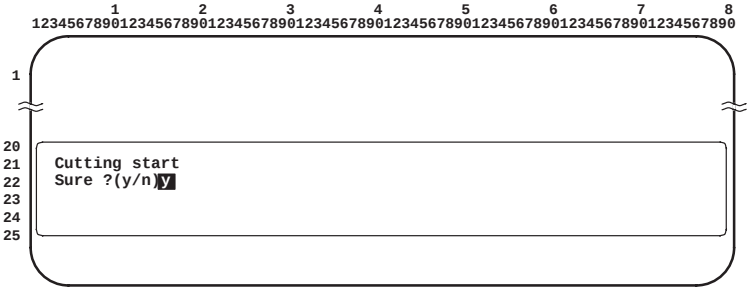
- Enter the number (1 to 4) of a program to be changed.
- Enter the new file name, program number, and repetition count. If the return key is pressed without entering the information described above, no change occurs.

(4) Delete program command



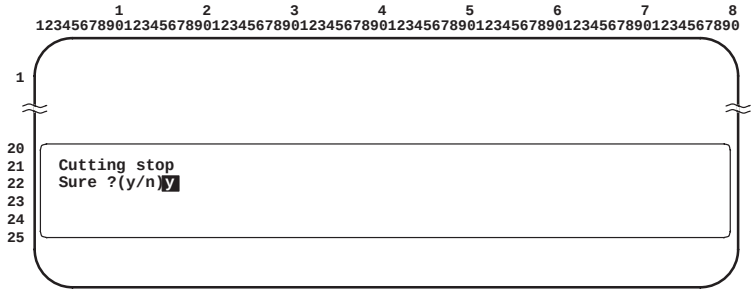
- A specified program is deleted from the registered-program display area. Any program names displayed below the deleted program name will not be shifted up.

(5) Start command



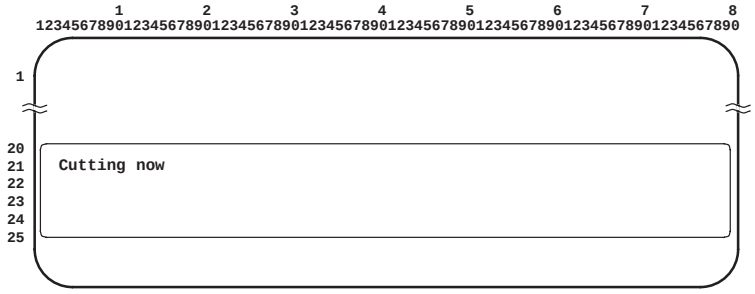
- Machining is started. Then, data is displayed in the current program data display area.
- Changes regarding the registered programs cannot be made during machining.

(6) Stop command



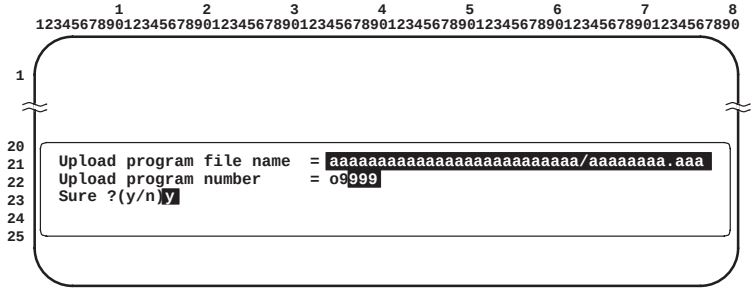
- Machining is stopped. Then, changes regarding the registered programs can be made.
- When the Start command is selected again, the lowest-numbered program is downloaded.

(7) Continue



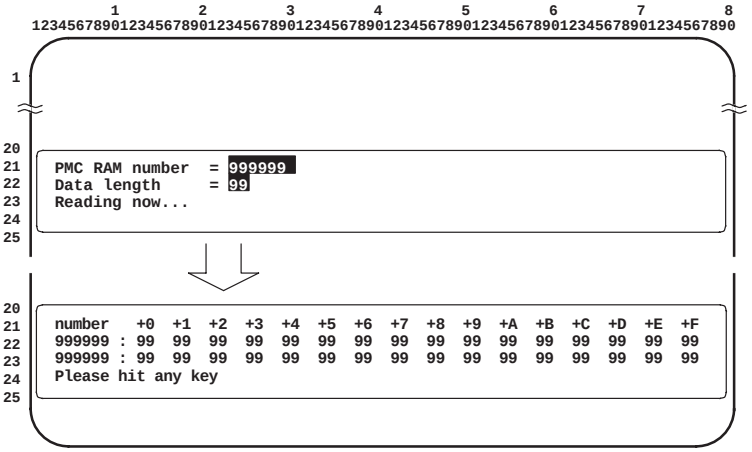
- Machining is resumed.
- After the ESC key is pressed to display the command entry menu and interrupt machining, this command is used to resume machining.

(8) Upload command



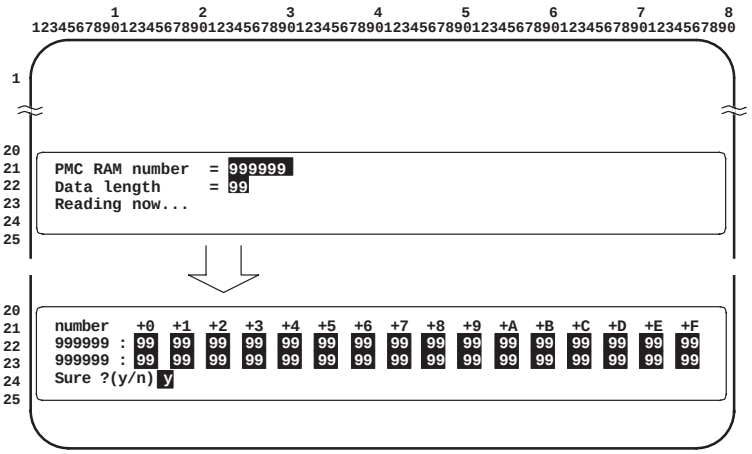
- Enter the name of a program file and the number of a program to be uploaded.
- Entering 0∗ in place of a program number, all programs are uploaded.

(9) Read PMC data command



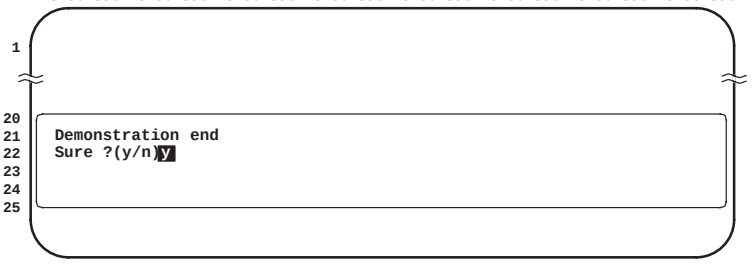
- Enter a PMC RAM number and data length, and the PMC data will be displayed.
- To resume the command entry menu, press any key.

(10) Write PMC data command



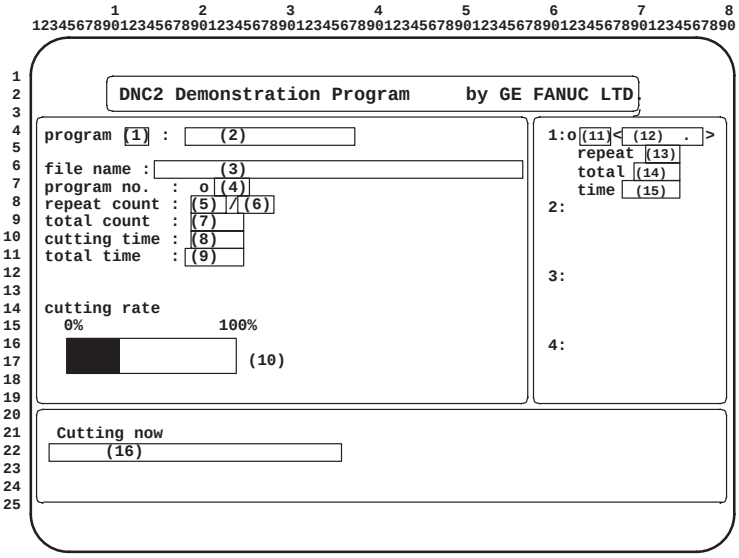
- Enter a PMC RAM number and data length, then PMC data.
- After all PMC data is entered, it is written to memory.

(11) End command



- The demonstration program is terminated, and control is returned to DOS.

K.4.3 Machining Menu



(1) Number of a program being executed (1 to 4)

(2) NC control command being executed (executing, downloading, deleting)

(3) Registered-program file name (fully qualified)

(4) Registered-program number

(5) Number of times the program has been executed (execution count)

(6) Specified repetition time

(7) Total execution count (since beginning of machining)

(8) Machining time (single pass) in hh:mm

(9) Total machining time in hh:mm

(10) Ratio of repeated times to specified repetition count

(11) Number of a program with registration number 1

(12) File name (with no path indicated) for a program with registration number 1

(13) Repetition count for a program with registration number 1

(14) Total execution count for a program with registration number 1

(15) Total machining time for a program with registration number 1

(16) Error indication (such as file open error or file read error); if an error occurs, machining is discontinued, and message HPleas hit any keyI is displayed. Pressing any key resumes the command entry menu.

L PMC ADDRESS (Series 15–A AND ADDRESS SPECIFICATION TYPE OF Series 15–B)

An address to be accessed is specified with an 8–digit hexadecimal number. (Example: For F000, specify 0X41FFE200.)

The accessible addresses are as follows:

PMC address	Byte address	Specified address	PMC ADDRESS	BIT ADDRESS
C000 G511	0 511	41FFE000 41FFE1FF	G0000.0 G0511.7	0 4095
F000 F319	512 831	41FFE200 41FFE33F	F0000.0 F0319.7	4096 6655
Y000 Y127	1024 1151	41FFE400 41FFE47F	Y0000.0 Y0127.7	8192 9215
X000 X127	1152 1279	41FFE480 41FFE4FF	X0000.0 X0127.7	9216 10239
A000 A024	1984 2008	41FFE7C0 41FFE7D8	A000.0 A024.7	15872 16071
R000 R499	2010 2509	41FFE7DA 41FFE9CD	R0000.0 R0499.7	16080 20079
R500 R999	2510 3009	41FFE9CE 41FFEBC1	R0500.0 R0999.7	20080 24079
T000 T079	3024 3103	41FFEBD0 41FFEC1F	T0000.0 T0079.7	24192 24831
K000 K018	3112 3130	41FFEC28 41FFEC3A	K0000.0 K0018.7	24896 25047
C000 C079	3132 3211	41FFEC3C 41FFEC8B	C0000.0 C0079.7	25056 25695
D000 D883	3212 4095	41FFEC8C 41FFEFFF	D0000.0 D0883.7	25696 32767
D0884 D1859	4096 5071	41FFF000 41FFF3CF	D0884.0 D1859.7	32768 40575

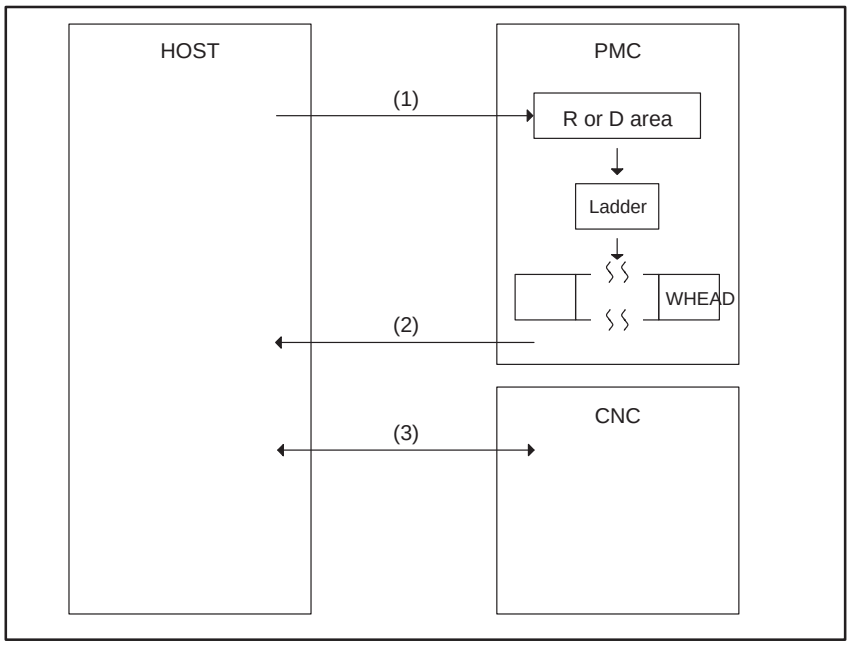
M DNC2 TOOL POST SELECTION (Series 15–TTB)

M.1 OUTLINE

When tool–post–dependent data is input/output with the DNC2 function of the Series 15–TTB, the target tool post can be selected.

This feature allows CNC data for tool posts 1 and 2 to be read and written.

M.2 ASWITCHING OF TARGET TOOL POST FOR INPUT/OUTPUT BY THE HOST COMPUTER



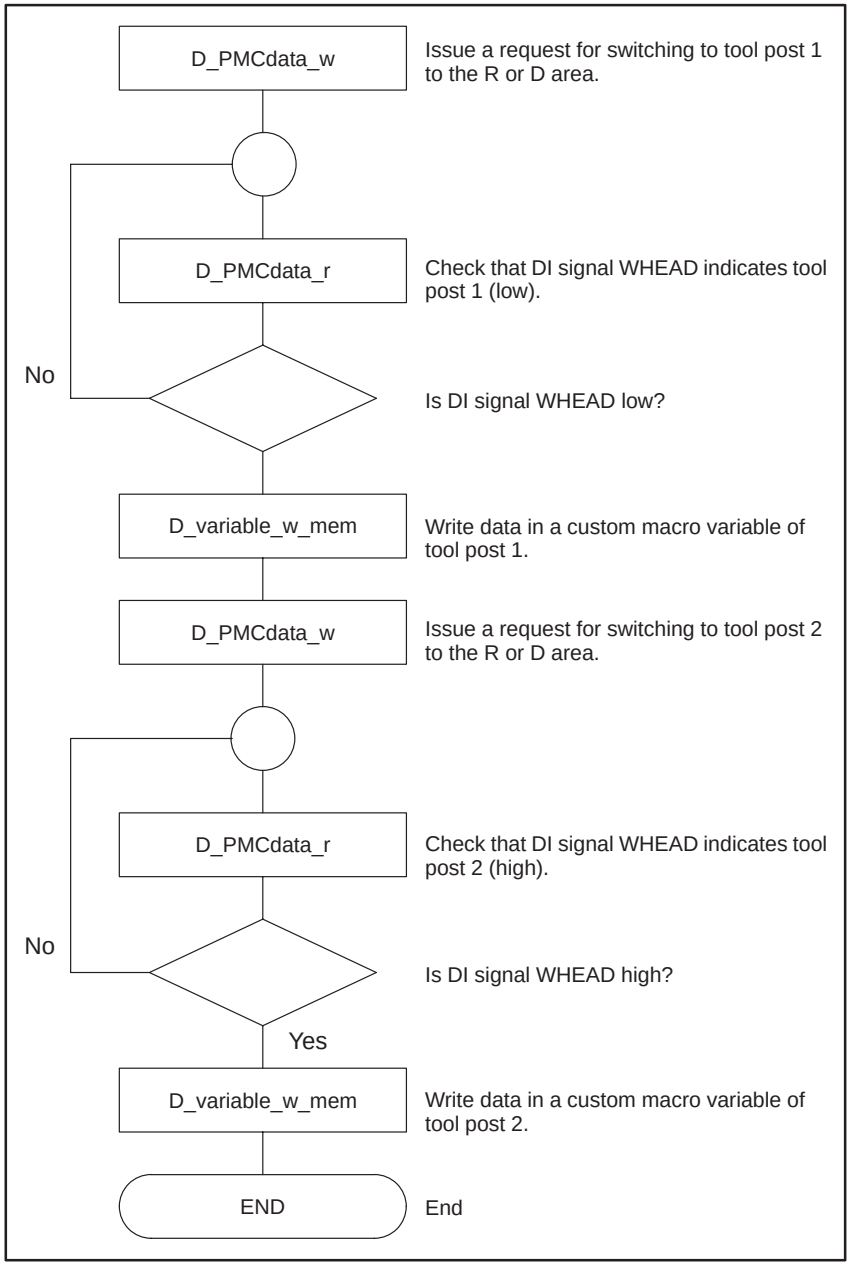
(1) Using the PMC data write function of DNC2, the host computer issues a request for switching to a target tool post for input/output to the R or D area.

(2) Using the PMC data read function of DNC2, the host computer confirms that DI signal WHEAD has been changed.

(3) Using these DNC2 functions, the host computer reads and writes tool–post–dependent data.

M.3 EXAMPLE

An outlined flow for writing data to tool posts 1 and 2 is shown below.



NOTE
1 All the names in the flowchart are function names of the DNC2 communication package.
2 For details of each function, see Section III–4.4.
3 Directly changing the status of DI signal WHEAD is not advised. The status of the signal should be changed by a ladder using, for example, an internal relay.

M.4 PARAMETERS

	#7	#6	#5	#4	#3	#2	#1	#0
2004				DIHD				

Data type : Bit

DIHD : When a DNC1 service function (main program selection, data read, or CNC data write) involving tool post specification, or a DNC1 service function (main program start) involving tool post selection is used:
0 : A tool post is specified or selected using a command from the host.
1 : A tool post is specified or selected using a DI signal.

When a DNC2 service function involving tool post selection is used:
0 : No tool post is selected.
1 : Tool post 1 or 2 is selected using a DI signal.

When using DNC2 with the Series 15–TTB, set this bit to 1.

M.5 CONNECTION

Tool post specification signal for NC window input/output **WHEAD**

[Type] Input signal

[Function] When a window function instruction or DNC2 function instruction is executed, this signal selects tool post 1 or 2.

[Operation] If this signal is low, executing a window function instruction or DNC2 function instruction causes input/output of the data related to tool post 1.

If this signal is high, executing a window function instruction or DNC2 function instruction causes input/output of the data related to tool post 2.

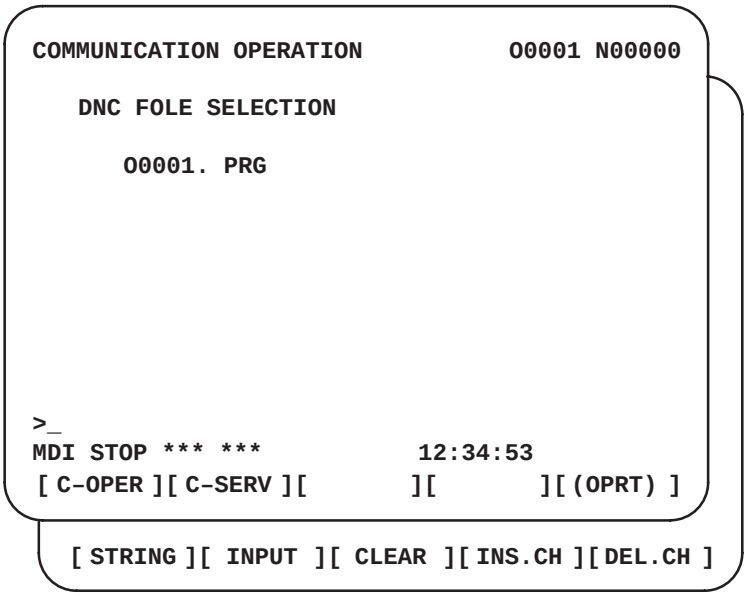
It takes at least 8 ms for the WHEAD signal to be accepted by the NC after the signal status is changed.

N DNC2 SCREEN (Series 16/18)

N.1 COMMUNICATION OPERATION SCREEN

When DNC operation is performed with the DNC2 function, the program name set on the communication operation screen is sent to the host.

This screen is displayed by using function key <SYSTEM> then the **[C–OPER]** soft key.



DNC file selection : Set the name of a program file to be executed by CNC cycle start in DNC operation mode. When CNC cycle start is performed in DNC operation mode (RMT), the program name set on this screen is sent to the host.

DNC file name format :
Oxxxx.PRG

xxxx : Program number (4 digits)

Example : When O0010.PRG is set, then a cycle start is performed, the following instruction is sent from the CNC to host:

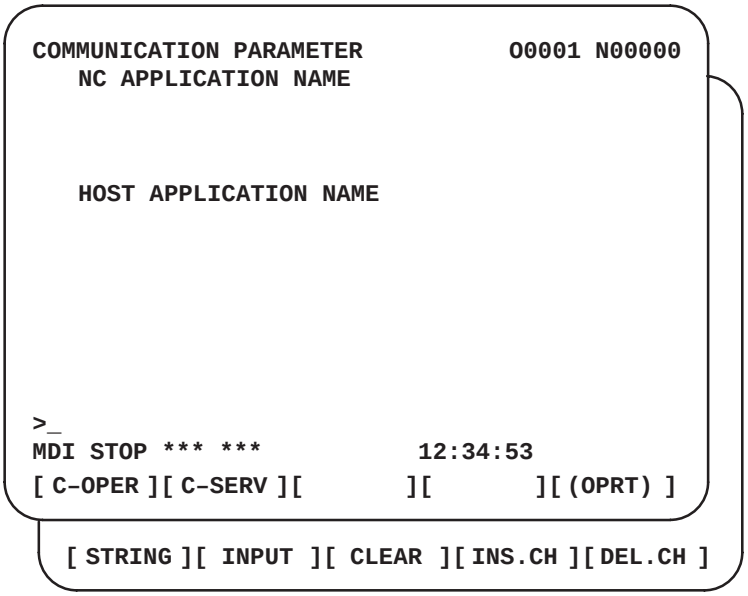
CNC Host
PTPM0010 ——>

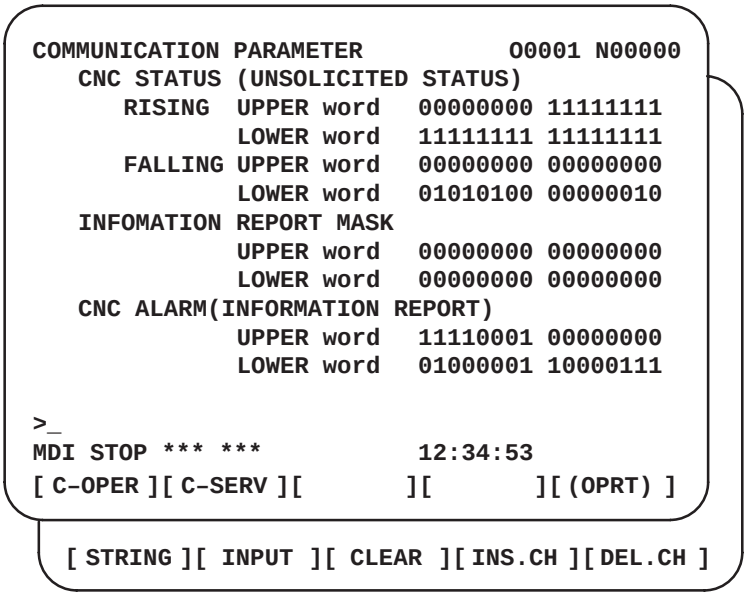
N.2 COMMUNICATION SERVICE SCREEN

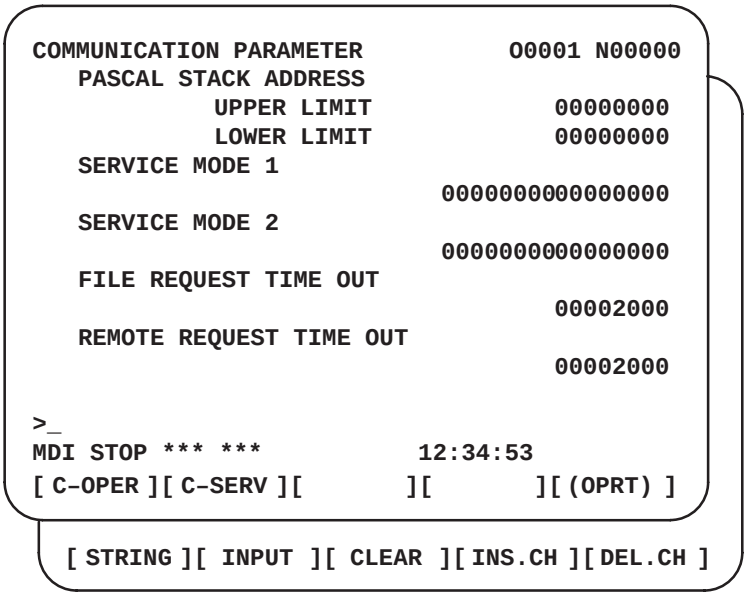
Before automatic notification of status/alarm information to the host can be performed with the DNC2 function, notification of information must be enabled by setting the DNC2 parameters on the communication service screen.

This screen is displayed by using function key <SYSTEM> then the *** F.46–4 *** soft key.

The communication service screen consists of the three pages. Use the page keys to switch between the pages described below.







The DNC2 function uses only part of the communication parameters. The parameters used are as follows:

a) NC application name : Unused
b) Host application name : Unused
c) Status notification enabled : Used
d) Status notification mask : Unused
e) Alarm notification : Used
f) Pascal stack address : Unused
g) Service mode 1 : Unused
h) Service mode 2 : Unused
i) File request timeout : Used
j) Remote request timeout : Used
k) Status notification enabled

c) Status notification enabled

This parameter determines whether the host is notified when the CNC status changes.

Notification of changes in signal status (rise and fall) can be enabled or disabled.

Bit
0 : Rewind (RWD)
1 : Alarm (AL)
2 : Reset (RST)
3 : Automatic operation stopped (SPL)
4 : Automatic operation being started (STL)
5 : Automatic operation in progress (OP)
6 : Servo ready (SA)
7 : CNC ready (MA)
8 : Unused
9 : Unused
10: Unused
11: Unused
12: M00 output (M00)
13: M01 output (M01)
14: M02 output (M02)
15: M30 output (M30)

e) Alarm notification

This parameter sets whether to notify the host if an alarm is generated in the CNC.

Bit
0 : Foreground PS alarm
1 : Background PS alarm
2 : Overheat alarm
3 : Undefined
4 : Undefined
5 : Parameter input enable alarm
6 : Overtravel (OT) alarm
7 : PMC error
8 : External alarm
9 : Undefined
10: Severe PS alarm
11: Undefined
12: Servo alarm
13: I/O alarm
14: Undefined
15: Battery alarm

i) File request timeout

This parameter sets the timeout value used for the time from when a file request is issued from the host to CNC until the CNC accepts the request.

If a timeout occurs, the CNC returns an error (M_NP 0XFDFF) to the host.

NOTE
1 A file request is a request for program downloading or uploading.
2 The parameter value is set as follows:
Unit : ms
Range : 16 ms to 9999.9984 s
(If 0 is set, 26 s is assumed.)

j) Remote request timeout

This parameter sets the timeout value used for the time from when a remote request is issued from the host to CNC until the CNC accepts the request.

If a timeout occurs, the CNC returns an error (M_NP 0XFDFF) to the host.

NOTE
1 A remote request is a request from the host to the CNC except program downloading and uploading requests.
2 The parameter value is set as follows:
Unit : ms
Range : 16 ms to 9999.9984 s
(If 0 is set, 26 s is assumed.)

Index

[M]

[N]

[O]

[P]

[R]

[S]

[T]

[U]

[V]

[W]

Revision Record

FANUC DNC 2 DESCRIPTIONS (B–61992E)

Edition	Date	Contents	Edition	Date	Contents
02	Oct., ’97	• Addition of Series 16–A/B/C • Addition of Series 16*i*/18*i*/21*i*–A			
01	Nov., ’91				