BNP-B2199* (ENG)

MITSUBISHI CNC

MELDAS MAGIC 64

CUSTOM APPLICATION INTERFACE LIBRARY GUIDE (VARIABLE SECTION)

ADVANCED AND EVER ADVANCING
MITSUBISHI ELECTRIC

Introduction

This instruction manual describes the Custom Application Interface Library (Custom API Library) used for developing the custom applications of MELDASMAGIC Series.
This instruction manual describes the Custom API variables used in developing the custom applications. Before programming, read this instruction manual thoroughly along with the instruction manuals shown below.

Custom Application Interface Library Guide

(Programming Section) BNP-B2197
(Function Section)..... BNP-B2198

Please read the following "Precautions for Safety" to ensure safe use of the MELDASMAGIC Series.

Precautions for Safety

Always read the specifications issued by the machine maker, this manual, related manuals and enclosed documents before starting installation, operation, programming, maintenance or inspections to ensure correct use. Thoroughly understand the basics, safety information and precautions of this numerical controller before using the unit.
The safety precautions are ranked as "DANGER", "WARNING" and "CAUTION" in this manual.

⚠ **DANGER**	When there is a great risk that the user could be subject to fatalities or serious injuries if handling is mistaken.
⚠ **WARNING**	When the user could be subject to fatalities or serious injuries if handling is mistaken.
⚠ **CAUTION**	When the user could be subject to injuries or when physical damage could occur if handling is mistaken.

Note that even if the item is ranked as " ⚠ CAUTION", incorrect handling could lead to serious results. Important information is described in all cases, so please observe the items.

⚠ **DANGER**
Not applicable in this manual.

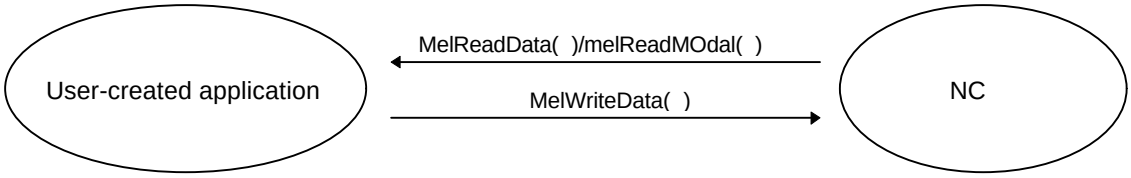
⚠ **WARNING**
Not applicable in this manual.

⚠ **CAUTION**
Items related to product and manual ⚠ For items described as "Restrictions" and "Usable State" etc., the instruction manual issued by the maker takes precedence over this manual. ⚠ Items not described in this manual must be interpreted as "Not Possible". ⚠ This instruction manual has been written on the assumption that all options are provided. Check the specifications issued by the machine maker before starting use. ⚠ Some screens and functions may differ or may not be used depending on the NC system version.

Contents

1. Outline

Because MELDASMAGIC reads and writes various information internally in the NC, it uses Custom Application Interface functions (melReadData, melWriteData, melReadModal). This manual describes information about the arguments for the Custom Application Interface function (hereafter, API function).



For more details about the API function, refer to the Custom Application Interface Library instruction manual, function section (BNP-B2142). For more details about how to use the API function, refer to the Custom Application Interface Library Guide (Programming Section) (BNP-B2141).

The various NC information is typed as shown below in the manual.

<Data type>

1. Data related to execution state of machining program, etc.
2. Data related to M/S/T/B functions
3. Data related to variable (#_____)
4. Data related to servo axis (name, position, state, etc.)
5. Data related to date and time, etc.
6. Data related to workpiece offset
7. Data related to tool offset (M/D)
8. Data related to tool offset (L/G)
9. Data related to tool life monitoring (M/D)
10. Data related to tool life monitoring (L/G)
11. Data related to ATC tool registration
12. Data related to alarms
13. Data related to servo axis operation state, etc.
14. Data related to spindle operation status
15. Data related to machine specifications
16. Data related to NC serial input/output
17. Data related to NC specification tuning
18. Data related to NC system
19. PLC device

2. How to Read the API Variable List

This chapter describes how to read the next chapter API variable list and the necessary points of programming.
In the API variable list in the next chapter, the necessary information when accessing (reading/writing) the data in the NC (section Nos./sub-section Nos., system and axis designation necessity, units for reading/writing, etc.) is arranged in the format below.

(1) Data title

(2) Detailed explanation of data

(3) Necessity of system designation (The system must be designated for circled data)
Necessity of axis designation (The axis must be designated for circled data)

(4) Values designating read/write data (section Nos./sub-section Nos.)

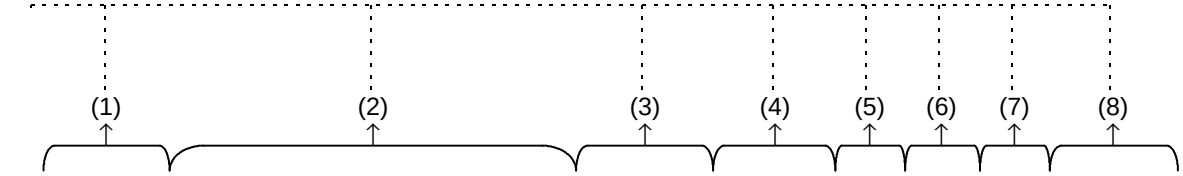
(5) Shows the necessary data size when reading NC data or writing data.
In the NC the data is stored in this data size.
The descriptions "1/4" and "2/X" are shown in the "Size/Bit" column. "1/4" means 1-byte of data, and shows that the information relevant to the fourth bit is included. "2/X" means two bytes of data, and shows that it has the meaning (for example, the first bit is 1st axis) of the bit position. For data not handled in bits, only the data size is shown in byte units.

(6) Validity of writing (Do not write data having "not possible" in the "Write" column)

(7) Units for reading/writing data
When indicated as 1=0.5[μm], if the retrieved value is "2000", it will be handled as 2000 × 0.5 [μm] = 1.000[mm]. (for 1μm, millimeter system)
Units will change depending on whether the NC system is in inches or millimeters, and depending on the input units. (Refer to sections 17.2 and 17.3)

	#1003	A (10μm)	B (1μm)	C (0.1μm)
#1041	Millimeter	1 = 5 [μm]	1 = 0.5 [μm]	1 = 0.05 [μm]
	Inch	1 = 1/2000 [inch]	1 = 1/20000 [inch]	1 = 1/200000 [inch]

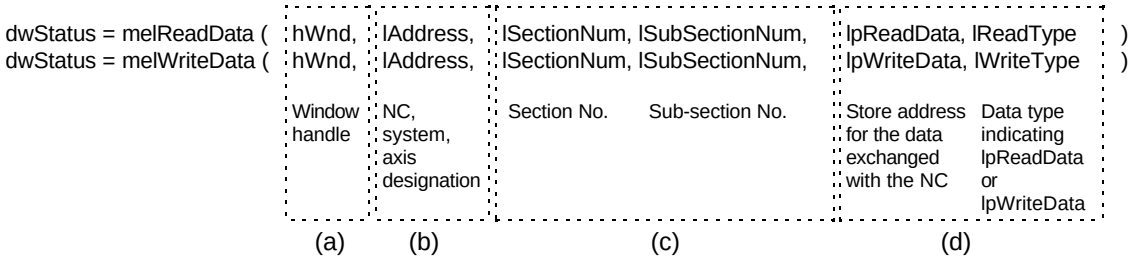
(8) Valid range of reading/writing data (Do not set a value other than this.)



	Data type				Custom API variable			Default data type			
	Item	Details	System designa-tion	Axis designa-tion	Section No.	Sub-section No.	Size/Bit	Data type symbol	Write	Unit	Data range
1. Data related to execution state of machining program, etc.											
1.1	0: Program No. (Main program)	No. of main program currently searched or being automatically run.	○	–	13	40031	4	T_LONG	Not possible	-	1~99999999
1.2	N: Sequence No. (Main program)	Sequence No. of main program currently searched or being automatically run.	○	–	13	40032	4	T_LONG	Not possible	-	0~99999
1.3	B: Block No. (Main program)	Block No. of main program currently searched or being automatically run.	○	–	13	40033	1	T_CHAR	Not possible	-	0~99

2.1 API function arguments

It is necessary to designate six arguments for the API functions melReadData() and melWriteData(). Those arguments are sorted into four categories (a)~(d) below and explained. (melReadModal() can also be used instead of melReadData(), but the explanation here uses melReadData() as an example. Refer to the "Custom Application Interface Library instruction manual, function section".)



(a) Designate the window handle in the hWnd argument of the melReadData and melWriteData(). For Visual Basic, this is accomplished by normally describing Me.hWnd. When using Windows API from Visual C/C++, etc., the CreateWindow() function return value is the window handle. Refer to the manual in the relevant language for more details.

(b) For the argument lAddress, designate the NC Card, and then designate the axis No. and system having the data to be read/written. The system designation and axis designation have necessary and unnecessary data. Refer to the list (system designation, axis designation) in the next chapter for more information about the system designation and axis designation in each data. For details on lAddress, refer to "Chapter 1.2 Address designation" of the "Custom Application Interface Library Guide (Function Section)".

(c) The argument lSectionNum, lSubsectionNum is a number designating data to be read/written. All data is given unique numbers. Consult the next chapter for the data to be read/written, and designate the section No. of that data in the argument lSectionNum, and the sub-section No. of that data in the argument lSubSectionNum.

(d) In the arguments lpReadData, lpWriteData, designate the storage places (addresses) of the data to be exchanged with the NC. When reading data (lReadData), this address becomes the designation of the area where the data received from the NC is stored. When writing data (lWriteData), this address becomes the designation of the place where the data to be transmitted to the NC was stored. Although data reading and data writing functions differ, the data structure is the same. Variables such as 1-byte/2-byte/4-byte/integer-type, 8-byte real number type, binary/decimal/hexadecimal character string type or real number character string type, etc., can be used. (Refer to Table 2.1)
For the argument lReadType, lWriteType, designate the data type (required data type) that argument lpReadData, lpWriteData indicates. For example, designate T_SHORT when 2-byte integer type data was designated in argument lpReadData. Designate T_LONG when 4-byte integer type data was designated in argument lpReadData. Designate T_FLOATSTR when real number character string type data was designated in argument lpReadData. (Refer to Table 2.2)

Using Table 2.1 and the "Default Data Types" list in the next chapter, determine the argument lpReadData, lpWriteData, and lReadType, lWriteType of the API functions melReadData() and melWriteData().

First find the target variable in the list in the next chapter, then find the "Default Data Type" of that variable.
Next, using Table 2.1, determine the data type most suitable to the application from the data types marked with a ○ in the default data type lines of the of the target variable. For example, for an application that displays the counter value with a character string, select a real number character string type data. To read the tool data into the list, select an integer type data if looking for the number of tool groups. (The data type to be handled by the application is called the required data type in this manual. Against this, the data type handled in the NC is called the default data. The API function automatically converts between the default data type and the required data type. Refer to Fig. 2.2)
Do not use any required data type in Table 2.1 having an × or △ mark attached. That data cannot be used or has usage restrictions.
After determining the required data type, find the API function argument and variable type declaration in Table 2.2.

Fig. 2.1 Variable list and API function argument and relation

	Data type				Custom API variable			Default data type			
	Item	Details	System designa-tion	Axis designa-tion	Section No.	Sub-section No.	Size/Bit	Data type symbol	Write	Unit	Data range
1. Data related to execution state of machining program, etc.											
1.1	0: Program No. (Main program)	No. of main program currently searched or being automatically run.	○	–	13	40031	4	T_LONG	Not possible	-	1~99999999
1.2	N: Sequence No. (Main program)	Sequence No. of main program currently searched or being automatically run.	○	–	13	40032	4	T_LONG	Not possible	-	0~99999
1.3	B: Block No. (Main program)	Block No. of main program currently searched or being automatically run.	○	–	13	40033	1	T_CHAR	Not possible	-	0~99

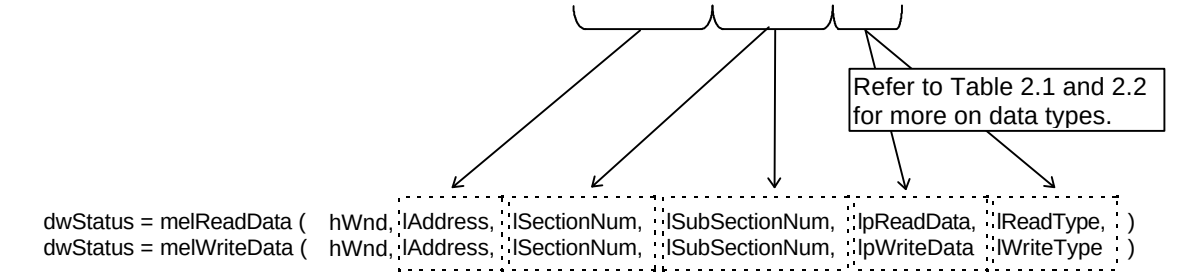


Fig. 2.2 Conversion of API function data type

The API function melReadData() converts the NC internal data to the data type designated by the argument lReadType, and stores it in the variable indicated by the argument lpReadData.
The API function melWriteData() carries out the opposite operation.

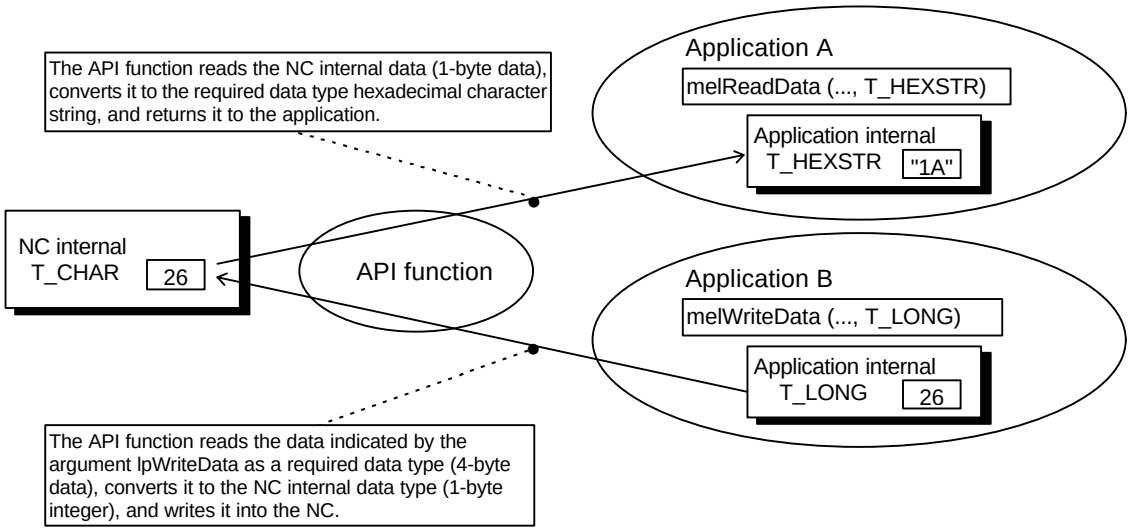


Table 2.1 Correspondence of default data types and required data types

	Required data type								
Default data type	**1-byte integer type**	**2-byte integer type**	**4-byte integer type**	**8-byte real number type**	**Character string type**	**Decimal integer character string type**	**Hexadecimal character string type**	**Binary character string type**	**Real number character string type**
T_BIT	○	○	○	○	×	○	○	○	○
T_CHAR	○	○	○	○	×	○	○	○	○
T_SHORT	△	○	○	○	×	○	○	○	○
T_LONG	△	△	○	○	×	○	○	○	○
T_DOUBLE	△	△	△	○	×	△	△	△	○
T_STR	×	×	×	×	○	×	×	×	×

Table 2.2 Correspondence of required data type and variable type declaration, API function argument

	Visual Basic example			**C/C++ example**		
Required data type	**Variable type declaration**	**lpReadData lpWriteData**	**lReadType lWriteType**	**Variable type declaration**	**lpReadData lpWriteData**	**lReadType lWriteType**
1-byte integer type	Dim c As Byte **(Note 1)**	c	T_CHAR	char c;	&c	T_CHAR
2-byte integer type	Dim n As Integer	n	T_SHORT	short n;	&n	T_SHORT
4-byte integer type	Dim l As Long	l	T_LONG	long l;	&l	T_LONG
8-byte real number type	Dim d As Double	d	T_DOUBLE	double d;	&d	T_DOUBLE
Character string type	Dim st As STRINGTYPE Dim s As String * 256	st	T_STR	STRINGTYPE st; char s[256];	&st	T_STR
Decimal integer character string type	Dim ds As STRINGTYPE Dim s As String * 256	ds	T_DECSTR	STRINGTYPE ds; char s[256];	&ds	T_DECSTR
Hexadecimal character string type	Dim hs As STRINGTYPE Dim s As String * 256	hs	T_HEXSTR	STRINGTYPE hs; char s[256];	&hs	T_HEXSTR
Binary character string type	Dim bs As STRINGTYPE Dim s As String * 256	bs	T_BINSTR	STRINGTYPE bs; char s[256];	&bs	T_BINSTR
Real number character string type	Dim fs As FLOATSTR Dim s As String * 256	fs	T_FLOATSTR	FLOATSTR fs; char s[256];	&fs	T_FLOATSTR

(Note 1) Byte designation cannot be used in Visual Basic 2.0 and 3.0. Declare the variable as Integer. Use T_SHORT in lReadType (or lWriteType).

(Note 2) The variable names in Table 2.2 are examples. Change the names to names with other meanings depending on necessity.

2.2 STRINGTYPE, FLOATSTR

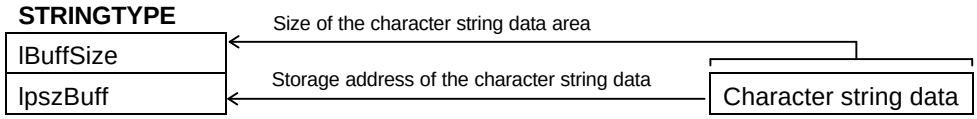
STRINGTYPE and FLOATSTR are data types defined for use in MELDASMAGIC. These types are defined in the following files.

melpcnc\sdk32\include\vb\meltype.bas (For Visual Basic)
melpcnc\sdk32\include\c\meltype.h (For C/C++)

[Using the files above]

- For Visual Basic, use the [File Add] in the [File] menu, integrate into the project, and then use.
- For C/C++, integrate these data types with #include at the head of the file to be used, and then use.

STRINGTYPE and FLOATSTR are defined respectively by the following structures.



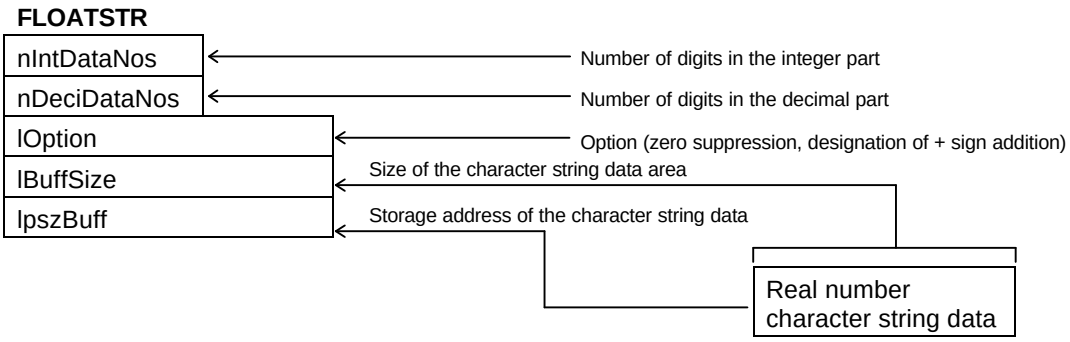
The following declaration and variable initialization are necessary when using STRINGTYPE.

<Visual Basic 2.0/3.0 example>

```
Dim st As STRINGTYPE
Dim s As String * 256

st.BuffSize = LenB(s)
dwStatus=melSetPointer(Me.hWnd, st.lpszBuff, s)
```

<C/C++ example>

```
STRINGTYPE  st;
char s[256];

st.Buffsize = sizeof(s);
st.lpszBuff = &s[0];
```



The following declaration and variable initialization are necessary when using FLOATSTR.

<Visual Basic 2.0/3.0 example>

```
Dim fs As FLOATSTR
Dim s As String * 256

fs.nIntDataNos = 5          'Number of digits in the
                            integer part
fs.nDECIDATANos = 3         'Number of digits in the
                            decimal part
fs.lOption = FLTSTR_DECI_ZERO_SUPPRESS
                            'Zero suppression
                            designation
fs.BuffSize = LenB(s)
dwStatus=melSetPointer(Me.hWnd, fs.lpszBuff, s)
```

<C/C++ example>

```
FLOATSTR  fs;
char s[256];

fs.nIntDataNos = 5;         /*Number of digits in the
                            integer part*/
fs.nDECIDATANos = 3;        /*Number of digits in the
                            decimal part*/
fs.lOption = FLTSTR_DECI_ZERO_SUPPRESS;
                            /*Zero suppression
                            designation*/
fs.BuffSize = sizeof(s);
fs.lpszBuff = &s[0];
```

3. List of API Variables

	Data type				Custom API variable			Default data type				
	Item	Details	System designation	Axis designation	Section No.	Sub-section No.	Size/ Bit	Data type symbol	Write	Unit	Data range	Remarks
1. Data related to execution state of machining program, etc.												
1.1	O: Program No. (Main program)	No. of main program currently searched or being automatically run.	O	—	13	40031	4	T_LONG	Not possible	—	1~99999999	
1.2	N: Sequence No. (Main program)	Sequence No. of main program currently searched or being automatically run.	O	—	13	40032	4	T_LONG	Not possible	—	0~99999	
1.3	B: Block No. (Main program)	Block No. of main program currently searched or being automatically run.	O	—	13	40033	1	T_CHAR	Not possible	—	0~99	
1.4	O: Program No. (Subprogram)	No. of subprogram being executed.	O	—	13	40034	4	T_LONG	Not possible	—	1~99999999	
1.5	N: Sequence No. (Subprogram)	Sequence No. of subprogram being executed.	O	—	13	40035	4	T_LONG	Not possible	—	0~99999	
1.6	B: Block No. (Subprogram)	Block No. of subprogram being executed.	O	—	13	40036	1	T_CHAR	Not possible	—	0~99	
1.7												
1.8												
1.9												
1.10	Subprogram call level	The nesting level (0 to 4) of the subprogram call.	O	—	13	16	1	T_CHAR	Not possible	—	0~4	Refer to section 3.1 for Macro call levels.
1.11	G command value (Group 1 command)	Group 1 (interpolation mode) G00, G01, G02, G03, G33 G command modal currently being executed.	O	—	13	40001	8	T_DOUBLE	Not possible	G command value 2=G02	0,1,2,3,33	When the L/G G code system 2 is selected, G32 is commanded instead of G33.
1.12	G command value (Group 2 command)	Group 2 (plane selection) G17, G18, G19 G command modal currently being executed.	O	—	13	40002	8	T_DOUBLE	Not possible	G command value 17=G17	17,18,19	
1.13	G command value (Group 3 command)	Group 3 (absolute) G90, (incremental) G91 G command modal currently being executed.	O	—	13	40003	8	T_DOUBLE	Not possible	G command value 90=G90	90,91	

	Data type				Custom API variable			Default data type				
	Item	Details	System designation	Axis designation	Section No.	Sub-section No.	Size/ Bit	Data type symbol	Write	Unit	Data range	Remarks
1.14	G command value (Group 4 command)	Group 4 (chuck barrier) G22, G23 G command modal currently being executed.	O	—	13	40004	8	T_DOUBLE	Not possible	G command value	22,23	L/G only
1.15	G command value (Group 5 command)	Group 5 (feed mode) G94, G95 G command modal currently being executed.	O	—	13	40005	8	T_DOUBLE	Not possible	G command value	94,95	When the L/G G code system 2 is selected, G98 and G88 are used.
1.16	G command value (Group 6 command)	Group 6 (inch) G20, (metric) G21 G command modal currently being executed.	O	—	13	40006	8	T_DOUBLE	Not possible	G command value 20=G20	20,21	
1.17	G command value (Group 7 command)	Group 7 (diameter offset mode) G40, G41, G42 G command modal currently being executed.	O	—	13	40007	8	T_DOUBLE	Not possible	G command value 41=G41	40,41,42	Including G46 for L/G
1.18	G command value (Group 8 command)	Group 8 (length offset mode) G43, G44, G49 G command modal currently being executed.	O	—	13	40008	8	T_DOUBLE	Not possible	G command value 43=G43	43,44,49	M/D only
1.19	G command value (Group 9 command)	Group 9 (fixed cycle mode) G70, G71, G72, G73, G74, G75, G76, G77, G78, G79, G80, G81, G82, G83, G84, G85, G86, G87, G88, G89 G command modal currently being executed.	O	—	13	40009	8	T_DOUBLE	Not possible	G command value 80=G80	70~89	When the L/G G code system 2 is selected, G90, G92 and G94 are also retrieved.
1.20	G command value (Group 10 command)	Group 10 (initial point return) G98, (R point return) G99 G command modal currently being executed.	O	—	13	40010	8	T_DOUBLE	Not possible	G command value 98=G98	98,99	
1.21	G command value (Group 11 command)	Group 11 G50.2, G51.2 G command modal currently being executed.	O	—	13	40011	8	T_DOUBLE	Not possible	G command value	50.2,51.2	L/G only
1.22	G command value (Group 12 command)	Group 12 (workpiece coordinate system modal) G54, G55, G56, G57, G58, G59 G command modal currently being executed.	O	—	13	40012	8	T_DOUBLE	Not possible	G command value 54=G54	54~59	
1.23	G command value (Group 13 command)	Group 13 (cutting mode) G61, G62, G63, G64 G command modal currently being executed.	O	—	13	40013	8	T_DOUBLE	Not possible	G command value 63=G63	61~64	
1.24	G command value (Group 14 command)	Group 14 (modal call) G66, G66.1, G67 G command modal currently being executed.	O	—	13	40014	8	T_DOUBLE	Not possible	G command value 66.1=G66.1	66,66.1,67	

	Data type				Custom API variable			Default data type				
	Item	Details	System designation	Axis designation	Section No.	Sub-section No.	Size/ Bit	Data type symbol	Write	Unit	Data range	Remarks
1.25												
1.26												
1.27	G command value (Group 17 command)	Group 17 (constant surface speed control) G96, G97 G command modal currently being executed.	O	—	13	40017	8	T_DOUBLE	Not possible	G command value	96,97	
1.28	G command value (Group 18 command)	Group 18 (balanced cut) G14, G15 G command modal currently being executed.	O	—	13	40018	8	T_DOUBLE	Not possible	G command value	14,15	L/G only
1.29	G command value (Group 19 command)	Group 19 (G command mirror image) G50.1, G51.1 G command modal currently being executed.	O	—	13	40019	8	T_DOUBLE	Not possible	G command value 50.1=G50.1	50.1,51.1	M/D only
1.30	G command value (Group 20 command)	Group 20 (spindle selection) G43.1, G44.1 G command modal currently being executed.	O	—	13	40020	8	T_DOUBLE	Not possible	G command value	43.1,44.1	
1.31												
1.32	Shape offset No. (D command value)	The tool diameter offset, D command value.	O	—	13	76	2	T_SHORT	Not possible	1=D1	1~200	The range follows the number of tool offset groups.
1.33	Wear offset No. (D command value)	The tool diameter offset, D command value.	O	—	13	78	2	T_SHORT	Not possible	1=D1	1~200	The range follows the number of tool offset groups.
1.34	Length offset No. (H command value)	The tool length offset, H command value.	O	O	14	112	2	T_SHORT	Not possible	1=H1	1~200	The range follows the number of tool offset groups.
1.35	FA: F command feedrate	Program command F modal value current being executed.	O	—	13	20224	8	T_DOUBLE	Not possible	1.=1[mm/min]	0.000 ~ 240000.000 [mm/min]	
1.36	FM: Manual effective feedrate	Manual feedrate (effective speed in axis direction)	O	—	20	20088	8	T_DOUBLE	Not possible	1.=1[mm/min]	0.000 ~ 240000.000 [mm/min]	
1.37	FS: Synchronous feedrate	Synchronous feedrate currently being executed.	O	—	11	40001	8	T_DOUBLE	Not possible	1.=1[mm/min]	[mm/rev]	

	Data type				Custom API variable			Default data type				
	Item	Details	System designation	Axis designation	Section No.	Sub-section No.	Size/ Bit	Data type symbol	Write	Unit	Data range	Remarks
1.38	Fc: Automatic effective feedrate	During interpolation feed: Vector direction speed currently being moved in. During independent feed of each axis: Speed of axis having highest speed.	O	—	11	40002	8	T_DOUBLE	Not possible	1.=1[mm/min]	0.000 ~ 240000.000 [mm/min]	
1.39	FE: Thread cutting	Lead value of thread cutting command currently being executed.	O	—	11	40003	8	T_DOUBLE	Not possible	1.=1[mm]	[mm]	
1.40	Dwell remaining time	Remaining dwell (G04) time 1/1000 (sec.)	O	—	11	20184	8	T_DOUBLE	Not possible	1.=1[sec.]	0.000 ~ 99999.999 [sec.]	
1.41	Invalid state: Single block	Macro variable #3003/bit 0 Block stop invalid	O	—	13	20	1/0	T_CHAR	Not possible	0=off/1=on	0,1	
1.42	Invalid state: MST complete	Macro variable #3003/bit 1 MST command completion signal wait	O	—	13	20	1/1	T_CHAR	Not possible	0=off/1=on	0,1	
1.43	Invalid state: Feed hold	Macro variable #3004/bit 0 Feed hold invalid	O	—	13	21	1/0	T_CHAR	Not possible	0=off/1=on	0,1	
1.44	Invalid state: Override	Macro variable #3004/ bit 1 Cutting override invalid	O	—	13	21	1/1	T_CHAR	Not possible	0=off/1=on	0,1	
1.45	Invalid state: Exact	Macro variable #3004/bit 2 G90 Block deceleration check invalid	O	—	13	21	1/2	T_CHAR	Not possible	0=off/1=on	0,1	
1.46	Automatic operation command	0: Positioning (axis independent) 1: Positioning (linear) 2: Linear interpolation 3: Circular interpolation (CW) 4: Circular interpolation (CCW) 5: Helical interpolation (CW) 6: Helical interpolation (CCW) 7: 8: 9: 10: 11: Time designated dwell 12: 13: 1st reference point check 14: 2nd reference point check 15: 3rd reference point check	O	—	11	71	1	T_CHAR	Not possible		0~29	

	Data type				Custom API variable			Default data type				
	Item	Details	System designation	Axis designation	Section No.	Sub-section No.	Size/ Bit	Data type symbol	Write	Unit	Data range	Remarks
1.46	Automatic operation command (continued)	16: 4th reference point check 17: Automatic reference point return 18: Return from automatic reference point 19: 2nd reference point return 20: 3rd reference point return 21: 4th reference point return 22: Skip function 23: Multi-step skip function 1 24: Multi-step skip function 2 25: Multi-step skip function 3 26: Thread cutting 27: 28: 29: Coordinate system setting	O	—	11	71	1	T_CHAR	Not possible		0~29	
1.47	In cutting mode	1 during G01, G02, G03, G31, G33, G34 and G35 modes, 0 during other modes.	0	—	11	20071	2	T_SHORT	Not possible	0=off/1=on	0, 1	
2. Data related to M/S/T/B functions												
2.1	M command	The miscellaneous function (M) command value. (Spindle start, stop, program stop, etc., designation)	—	—	10318	20($1) 220($2)	4	T_LONG	Not possible	BCD	0~99999999	$1: R[20/21] $2: R[220/221]
2.2	S command	The spindle speed (S) command value.	—	—	10318	8(SP1) 208(SP2)	4	T_LONG	Not possible		0~999999	SP1: R[8/9] SP2: R[208/209]
2.3	T command	Tool exchange (T) command value.	—	—	10318	36($1) 236($2)	4	T_LONG	Not possible	BCD	0~99999999	$1: R[36/37] $2: R[236/237]
2.4	B command	The secondary miscellaneous command value (designation of index table position, etc.).	—	—	10318	44($1) 244($2)	4	T_LONG	Not possible	BCD	0~99999999	$1: R[44/45] $2: R[244/245]
2.5	Manual numerical value command (M)	Manual numerical value command for miscellaneous function (M). This functions as a manual numerical value command by setting a numerical value.	O	—	20	40001	4	T_LONG	Possible		0~99999999	Reading not possible
2.6	Manual numerical value command (S)	Manual numerical value command for spindle rotation speed (S). This functions as a manual numerical value command by setting a numerical value.	—	O	20	40002	4	T_LONG	Possible		0~99999999	Reading not possible
2.7	Manual numerical value command (T)	Manual numerical value command for tool change (T). This functions as a manual numerical value command by setting a numerical value.	O	—	20	40003	4	T_LONG	Possible		0~99999999	Reading not possible
2.8												

	Data type				Custom API variable			Default data type				
	Item	Details	System designation	Axis designation	Section No.	Sub-section No.	Size/ Bit	Data type symbol	Write	Unit	Data range	Remarks
3. Data related to variable (#_____)												
3.1	Macro subprogram execution level	The subprogram call level (0 to 4) according to macro call (G65, G66, G66.1) is retrieved.	O	—	13	18	1	T_CHAR	Not possible		0~4	Refer to section 1.10 for call levels including M98 call.
3.2	Local variable macro subprogram: Level 0	The local variable #1 to #33 value when not in user macro call state is retrieved.	O	—	30	20000+ 1~33	8	T_DOUBLE	Not possible	1.0=1.0		
3.3	Local variable macro subprogram: Level 1	The local variable #1 to #33 value of user macro call level 1 is retrieved.	O	—	30	20000+ 101~133	8	T_DOUBLE	Not possible	1.0=1.0		
3.4	Local variable macro subprogram: Level 2	The local variable #1 to #33 value of user macro call level 2 is retrieved.	O	—	30	20000+ 201~233	8	T_DOUBLE	Not possible	1.0=1.0		
3.5	Local variable macro subprogram: Level 3	The local variable #1 to #33 value of user macro call level 3 is retrieved.	O	—	30	20000+ 301~333	8	T_DOUBLE	Not possible	1.0=1.0		
3.6	Local variable macro subprogram: Level 4	The local variable #1 to #33 value of user macro call level 4 is retrieved.	O	—	30	20000+ 401~433	8	T_DOUBLE	Not possible	1.0=1.0		
3.7	Number of common variable sets (#100~)	Number of common variable sets (#100~)	O	—	10	18220	1	T_CHAR	Not possible	40=40[sets]		
3.8	Common variable (#100~)	The common variables #100~ value. For these variables, the system needs to be designated. There are limits to the sub-section Nos. that can be used according to the number of common variable (#100~) sets specifications.	O	—	32	20000+ 100~199	8	T_DOUBLE	Possible	1.0=1.0		Refer to Chapter 4.1.
3.9	Number of common variable sets (#500~)	Number of common variable sets (#500~)	—	—	10	18222	1	T_CHAR	Not possible	40=40[sets]		
3.10	Common variable (#500~)	The common variables #500~ value. There are limits to the sub-section Nos. that can be used according to the number of common variable (#500~) sets specifications.	—	—	29	20000+ 500~599	8	T_DOUBLE	Possible	1.0=1.0		Refer to Chapter 4.1.
3.11	Common variable name	The names given to the common variables (#500 ~ #519).	—	—	29	40000+ 500~519	8	T_STR	Possible		7 alpha-numeric characters starting with an alphabet character.	7 characters + NULL
3.12	Variable <Open> #0	Value of the Variable (#1~#33, #100~#199, #500~#519) when the value is not set	—	—	29	20000	8	T_DOUBLE	Not possible			Refer to Chapter 4.1.

	Data type				Custom API variable			Default data type				
	Item	Details	System designation	Axis designation	Section No.	Sub-section No.	Size/ Bit	Data type symbol	Write	Unit	Data range	Remarks
4. Data related to servo axis (name, position, state, etc.)												
4.1	Number of Axes	The number of the servo axes for each system is obtained. Refer to the "Data related to tuning of NC specifications" for the number of systems.	—	—	1	513($1) 1537($2)	1	T_CHAR	Pro-hibited	1=1[axis]	1~4	
4.2	Axis name	The name of each axis (servo axis) is retrieved.	O	O	2	0	1	T_CHAR	Possible		X,Y,Z,U,V,W, A,B,C	
4.3	Axis reference return state #1	The axis that completed the 1st reference point return. The bit corresponding to the axis that manually or automatically returned to the reference point is set to "1". **Example)** 00000101(2)= 1st, 3rd axis arrived at the #1 zero point	—	—	23	8	2/X	T_SHORT	Not possible			
4.4	Axis reference return state #2	The axis that completed the 2nd reference point return. The bit corresponding to the axis that manually or automatically returned to the reference point is set to "1". **Example)** 00001010(2)= 2nd, 4th axis arrived at the #2 zero point	—	—	23	10	2/X	T_SHORT	Not possible			
4.5	Axis reference return state #3	The axis that completed the 3rd reference point return. The bit corresponding to the axis that manually or automatically returned to the reference point is set to "1". **Example)** 00000010(2)= 2nd axis arrived at the #3 zero point	—	—	23	12	2/X	T_SHORT	Not possible			
4.6	Axis reference return state #4	The axis that completed the 4th reference point return. The bit corresponding to the axis that manually or automatically returned to the reference point is set to "1". **Example)** 00001000(2)= 4th axis arrived at the #4 zero point	—	—	23	14	2/X	T_SHORT	Not possible			
4.7	Axis state (axis removed) ><	The axis is now removed.	O	O	27	41	1/4	T_CHAR	Not possible			When bit 4: ON, axis is being removed.
4.8	Axis state (servo OFF)][	The bit corresponding to the axis for which servo is OFF is set to [1].	—	—	23	0	2/X	T_SHORT	Not possible			

	Data type				Custom API variable			Default data type				
	Item	Details	System designation	Axis designation	Section No.	Sub-section No.	Size/ Bit	Data type symbol	Write	Unit	Data range	Remarks
4.9	Axis state (mirror image) MR	The bit corresponding to the axis in mirror image is set to [1].	O	—	13	84	2/X	T_SHORT	Not possible			
4.10	Machine position	Each axis coordinate position in basic machine coordinate system.	O	O	21	20000	8	T_DOUBLE	Not possible	1. = 1[mm]	–99999.999~ +99999.999 [mm]	
4.11	Current position	Incremental position from preset point after dog type zero point return or according to G92/origin set/counter set	O	O	21	20032	8	T_DOUBLE	Not possible	1.=1[mm]	–99999.999~ +99999.999 [mm]	
4.12	Workpiece coordinate position	Coordinate position in current workpiece coordinate system	O	O	21	20036	8	T_DOUBLE	Not possible	1.=1[mm]	–99999.999~ +99999.999 [mm]	
4.13	Remaining command	Remaining distance of movement command being executed.	O	O	12	20004	8	T_DOUBLE	Not possible	1.=1[mm]	–99999.999~ +99999.999 [mm]	
4.14	Next movement amount	The movement amount of the next block is retrieved.	O	O	16	20004	8	T_DOUBLE	Not possible	1.=1[mm]	–99999.999~ +99999.999 [mm]	
4.15	Origin set	By setting this to 0, the coordinate system is shifted so that the current machine position is the zero position of the part program coordinate system.	O	O	21	40002	8	T_DOUBLE	Possible	1.=1[mm]	–99999.999~ +99999.999 [mm]	Reading not possible
4.16	Counter set	By setting this to 0, the current position counter is preset to 0.	O	O	21	40001	8	T_DOUBLE	Possible	1.=1[mm]	–99999.999~ +99999.999 [mm]	Reading not possible
4.17	Interpolation vector length	Interpolation distance (Distance between start point and end point during movement with G01 command)	O	—	11	172	4	T_LONG	Not possible	1=0.5[µm]	–99999.999~ +99999.999 [mm]	
4.18	Manual interrupt amount (1)	Manual interrupt amount during manual ABS switch OFF	O	O	21	8	4	T_LONG	Not possible	1=0.5[µm]	–99999.999~ +99999.999 [mm]	When manual ABS switch is ON
4.19	Manual interrupt amount (2)	Manual interrupt amount during manual ABS switch ON	O	O	21	12	4	T_LONG	Not possible	1=0.5[µm]	–99999.999~ +99999.999 [mm]	When manual ABS switch is ON
4.20	Position at machine coordinate system during skip ON	The machine coordinate position at the skip ON point is retrieved.	O	O	21	116	4	T_LONG	Not possible	1=0.5[µm]	–99999.999~ +99999.999 [mm]	
4.21	Position at workpiece coordinate system during skip ON	The workpiece coordinate position at the skip ON point is retrieved.	O	O	21	112	4	T_LONG	Not possible	1=0.5[µm]	–99999.999~ +99999.999 [mm]	
4.22	Remaining distance during skip ON	The remaining distance at the skip ON point is retrieved.	O	O	21	120	4	T_LONG	Not possible	1=0.5[µm]	–99999.999~ +99999.999 [mm]	

	Data type				Custom API variable			Default data type				
	Item	Details	System designation	Axis designation	Section No.	Sub-section No.	Size/ Bit	Data type symbol	Write	Unit	Data range	Remarks
5. Data related to date and time, etc.												
5.1	Date	Date (Year, Month, Date) **Example)** 1995/12/02=951202	—	—	24	40001	4	T_LONG	Possible		000101 ~991231	
5.2	Time	NC built-in clock Hour, Min., Sec. **Example)** 23:59:59=235959	—	—	24	40002	4	T_LONG	Possible		0~235959	
5.3	Power ON time	Total time control unit power ON to OFF Hour, Min., Sec. **Example)** 9999:59:59=99995959	—	—	24	20000	4	T_LONG	Possible		0~99995959	
5.4	Automatic operation time	Total incremented time of each machining time from when the automatic start button is pressed in Memory, Tape, or MDI mode to M02/M30 or when the reset button is pressed. Hour, Min., Sec. **Example)** 9999:59:59=99995959	—	—	24	20004	4	T_LONG	Possible		0~99995959	
5.5	Automatic start time	Total automatic start time to completion from feed hold, block stop or reset. Hour, Min., Sec. **Example)** 23:59:59=235959	—	—	24	20008	4	T_LONG	Possible		0~99995959	
5.6	External cumulated 1 time	Time (1) controlled by PLC. Hour, Min., Sec. **Example)** 9999:59:59=99995959 (When the cumulated time display reaches the max. value (9999:59:59) the counting is stopped, and the max. value remains displayed.)	—	—	24	20012	4	T_LONG	Possible		0~99995959	Count when Y234:ON
5.7	External cumulated 2 time	Time (2) controlled by PLC. Hour, Min., Sec. **Example)** 9999:59:59=99995959	—	—	24	20016	4	T_LONG	Possible		0~99995959	Count when Y235:ON
6. Data related to workpiece offset												
6.1	G54 workpiece coordinate offset	The offset amount from the machine zero point to the G54 workpiece coordinate system zero point is set.	O	O	4	0	4	T_LONG	Possible	1=0.5[µm]	–99999.999~ +99999.999 [mm]	
6.2	G55 workpiece coordinate offset	The offset amount from the machine zero point to the G55 workpiece coordinate system zero point is set.	O	O	4	4	4	T_LONG	Possible	1=0.5[µm]	–99999.999~ +99999.999 [mm]	
6.3	G56 workpiece coordinate offset	The offset amount from the machine zero point to the G56 workpiece coordinate system zero point is set.	O	O	4	8	4	T_LONG	Possible	1=0.5[µm]	–99999.999~ +99999.999 [mm]	
6.4	G57 workpiece coordinate offset	The offset amount from the machine zero point to the G57 workpiece coordinate system zero point is set.	O	O	4	12	4	T_LONG	Possible	1=0.5[µm]	–99999.999~ +99999.999 [mm]	

	Data type				Custom API variable			Default data type				
	Item	Details	System designation	Axis designation	Section No.	Sub-section No.	Size/ Bit	Data type symbol	Write	Unit	Data range	Remarks
6.5	G58 workpiece coordinate offset	The offset amount from the machine zero point to the G58 workpiece coordinate system zero point is set.	O	O	4	16	4	T_LONG	Possible	1=0.5[μm]	–99999.999~ +99999.999 [mm]	
6.6	G59 workpiece coordinate offset	The offset amount from the machine zero point to the G59 workpiece coordinate system zero point is set.	O	O	4	20	4	T_LONG	Possible	1=0.5[μm]	–99999.999~ +99999.999 [mm]	
6.7	External workpiece coordinate system offset	The offset of the external workpiece coordinate system in which G54 to G59 are shifted is set.	O	O	4	24	4	T_LONG	Possible	1=0.5[μm]	–99999.999~ +99999.999 [mm]	
7. Data related to tool offset (50M/50D)												
7.1	Number of tool offset sets	Number of tool offset sets.	O	—	10	18208	2	T_SHORT	Not possible	200=200[set]	40, 80 100, 200	The number of sets depends on the NC specifications.
7.2	Tool offset type	Type of tool offset 1:M system type Ⅰ, 2:M system type Ⅱ, 6:L system type	O	—	10	18256	1	T_CHAR	Not possible		1, 4, 6	
7.3	Tool offset amount Only when M system type Ⅰ is selected	The tool offset amount is set or retrieved.	O	—	31	1~ Number of tool offset sets	4	T_LONG	Possible	1=0.5[μm]	–99999.999~ +99999.999 [mm]	
7.4	Tool offset amount: Length offset dimension Only when M system type Ⅱ is selected	The tool length offset amount (dimension) is set or retrieved.	O	—	31	1~ Number of tool offset sets	4	T_LONG	Possible	1=0.5[μm]	–99999.999~ +99999.999 [mm]	
7.5	Tool offset amount: Length offset wear amount Only when M system type Ⅱ is selected	The tool length offset amount (wear) is set or retrieved.	O	—	31	1001~ Number of tool offset sets	4	T_LONG	Possible	1=0.5[μm]	–99999.999~ +99999.999 [mm]	
7.6	Tool offset amount: Diameter offset dimension Only when M system type Ⅱ is selected	The tool diameter offset amount (dimension) is set or retrieved.	O	—	31	6001~ Number of tool offset sets	4	T_LONG	Possible	1=0.5[μm]	–99999.999~ +99999.999 [mm]	
7.7	Tool offset amount: Diameter offset wear amount Only when M system type Ⅱ is selected	The tool diameter offset amount (wear) is set or retrieved.	O	—	31	7001~ Number of tool offset sets	4	T_LONG	Possible	1=0.5[μm]	–99999.999~ +99999.999 [mm]	
7.8	Reference surface height (SURFACE)	Reference surface coordinate position for tool length measurement Ⅱ	O	O	2	664	4	T_LONG	Possible	1=0.5[μm]	–99999.999~ +99999.999 [mm]	

	Data type				Custom API variable			Default data type				
	Item	Details	System designation	Axis designation	Section No.	Sub-section No.	Size/ Bit	Data type symbol	Write	Unit	Data range	Remarks
8. Data related to tool offset (50L/50G)												
8.1	Number of tool offset sets	Number of tool offset sets.	—	—	10	18208	2	T_SHORT	Not possible	80=80[set]	0~80	The number of sets depends on the NC specifications.
8.2	Tool offset amount : nose wear amount : X (Only when L system is selected)	The axis nose wear amount set with #1026-base-I is set or retrieved.	O	—	31	1001~ Number of tool offset sets	4	T_LONG	Possible	1=0.5[µm]	–999.999 ~ +99.999 [mm]	
8.3	Tool offset amount : nose wear amount : Z (Only when L system is selected)	The axis nose wear amount set with #1028-base-K is set or retrieved.	O	—	31	5001~ Number of tool offset sets	4	T_LONG	Possible	1=0.5[µm]	–999.999 ~ +99.999 [mm]	
8.4	Tool offset amount : nose wear amount : C (Only when L system is selected)	The axis nose wear amount set with #1027-base-J is set or retrieved.	O	—	31	3001~ Number of tool offset sets	4	T_LONG	Possible	1=0.5[µm]	–999.999 ~ +99.999 [mm]	
8.5	Tool length : X (Only when L system is selected)	The axis tool length offset data set with #1026-base-I is set or retrieved.	O	—	31	1~ Number of tool offset sets	4	T_LONG	Possible	1=0.5[µm]	–99.999 ~ +99.999 [mm]	
8.6	Tool length : Z (Only when L system is selected)	The axis tool length offset data set with #1028-base-K is set or retrieved.	O	—	31	4001~ Number of tool offset sets	4	T_LONG	Possible	1=0.5[µm]	–99.999 ~ +99.999 [mm]	
8.7	Tool length : C (Only when L system is selected)	The axis tool length offset data set with #1027-base-J is set or retrieved.	O	—	31	2001~ Number of tool offset sets	4	T_LONG	Possible	1=0.5[µm]	–99.999 ~ +99.999 [mm]	
8.8	Tool nose radius : R (Only when L system is selected)	The nose R offset amount is set or retrieved.	O	—	31	6001~ Number of tool offset sets	4	T_LONG	Possible	1=0.5[µm]	0.000 ~ +99.999 [mm]	
8.9	Tool nose radius wear amount : r (Only when L system is selected)	The nose R wear offset amount is set or retrieved.	O	—	31	7001~ Number of tool offset sets	4	T_LONG	Possible	1=0.5[µm]	0.000 ~ +99.999 [mm]	
8.10	Hypothetical nose point No.: P (Only when L system is selected)	The hypothetical nose point No. is set or retrieved.	O	—	31	8001~ Number of tool offset sets	1	T_CHAR	Possible		0~8	
8.11												

	Data type				Custom API variable			Default data type				
	Item	Details	System designation	Axis designation	Section No.	Sub-section No.	Size/ Bit	Data type symbol	Write	Unit	Data range	Remarks
9. Data related to tool life monitoring (50M/50D)												
9.1												
10. Data related to tool life monitoring (50L/50G)												
10.1	Tool life monitoring method I/II	Selection of tool life monitoring method I or II. 1: Method I, 2: Method II	—	—	1	70	1	T_CHAR	Possible		1,2	
11. Data related to ATC tool registration												
11.1	ATC tool registration: Control parameters	Bit0: 0 = T4-digit Bit1: 0 =1 Start magazine, 1=0 Start magazine When creating a screen display, refer to Bit1, and control whether to display from magazine pot No. 1 or 0.	—	—	10218	2950	2	T_SHORT	Not possible	0=off/1=on	Bit0,1	
11.2	ATC tool registration: No. of magazine registered tools	The No. of tools registered in the magazine is indicated.	—	—	10218	2960	2	T_SHORT	Not possible	Binary	1~80	
11.3	ATC tool registration: Mount	The No. of the tool mounted is retrieved.	—	—	10218	2970	2	T_SHORT	Not possible	BCD	1~9999	
11.4	ATC tool registration: Wait 1	The wait 1 tool No. is retrieved.	—	—	10218	2971	2	T_SHORT	Not possible	BCD	1~9999	
11.5	ATC tool registration: Wait 2	The wait 2 tool No. is retrieved.	—	—	10218	2972	2	T_SHORT	Not possible	BCD	1~9999	
11.6	ATC tool registration: Wait 3	The wait 3 tool No. is retrieved.	—	—	10218	2973	2	T_SHORT	Not possible	BCD	1~9999	
11.7	ATC tool registration: Index	The wait 4 tool No. is retrieved.	—	—	10218	2974	2	T_SHORT	Not possible	BCD	1~9999	
11.8	ATC tool registration: Tool	The No. of the tool stored in the magazine pot is retrieved. The 1st magazine pot tool No. is 3000, the 2 and 3 tool Nos., etc., are 3001, 3002, etc.	—	—	10218	3000~3079	2	T_SHORT	Possible	BCD	1~9999	
11.9	ATC tool registration: D	The tool registration expansion (for machine maker expansion) data is retrieved. The 1st magazine D value is 3160, the 2 and 3 tool Nos., etc., are 3161, 3162, etc.	—	—	10218	3160~3239	2	T_SHORT	Possible	Binary	1~80	
11.10	ATC tool registration: AUX	This is the I/F used by the user PLC to process a random sequence.	—	—	10218	2998	2	T_SHORT	Possible	Binary	0~99	
12. Data related to alarms												
12.1		The alarm is retrieved using the function melGetCurrentAlarmMsg. (Refer to "Function Section")										

	Data type				Custom API variable			Default data type				
	Item	Details	System designation	Axis designation	Section No.	Sub-section No.	Size/ Bit	Data type symbol	Write	Unit	Data range	Remarks
13. Data related to servo axis operation state, etc.												
13.1	Speed	Actual speed of motor	O	O	27	312	2	T_SHORT	Not possible	1=1[rpm]	0~	
13.2	Load current	Motor current (Continuous current is calculated during stalling, and displayed)	O	O	27	328	2	T_SHORT	Not possible	1=1[%]	0~	
13.3	Amplifier display	7-segment LED display on driver	O	O	27	386	1	T_CHAR	Not possible		00~FF	
14. Data related to spindle operation status												
14.1	Spindle (SR,SF) speed	Actual speed of spindle motor including override	—	—	26	8992 (SP1) 7712 (SP2)	4	T_LONG	Not possible	1=1[rpm]	0~	
14.2	Load	Spindle motor load	—	—	26	8988 (SP1) 7708 (SP2)	2	T_SHORT	Not possible	1=1[%]	0~	
14.3	Amplifier display	7-segment LED of driver	—	—	26	9010 (SP1) 7730 8SP2)	1	T_CHAR	Not possible		00~FF	Must be HEX displayed
15. Data related to machine specifications												
15.1	1st reference point coordinates #2037 G53ofs	The coordinate position of the reference point return completion position when the machine reference position coordinates are 0. (The machine position will be set to this value with reference point return after the power is turned ON.)	O	O	2	272	4	T_LONG	Possible	1=0.5[µm]	–99999.999~ +99999.999 [mm]	
15.2	2nd reference point coordinates #2038 #2_rfp	Coordinate position determined with G30P2 command.	O	O	2	276	4	T_LONG	Possible	1=0.5[µm]	–99999.999~ +99999.999 [mm]	
15.3	3rd reference point coordinates #2039 #3_rfp	Coordinate position determined with G30P3 command.	O	O	2	280	4	T_LONG	Possible	1=0.5[µm]	–99999.999~ +99999.999 [mm]	
15.4	4th reference point coordinates #2040 #4_rfp	Coordinate position determined with G30P4 command.	O	O	2	284	4	T_LONG	Possible	1=0.5[µm]	–99999.999~ +99999.999 [mm]	
15.5	Reference point return direction (–) #2030 dir(–)	Sets which side of the near-point dog the reference point is on for reference point return. 0: Positive direction, 1: Negative direction	O	O	2	20	2/2	T_SHORT	Possible	0=off/1=on	0,1	
15.6	Reference point no axis #2031 noref	An axis not having a reference point is designated. 0: Normal control axis 1: Axis with no reference point	O	O	2	20	2/10	T_SHORT	Possible	0=off/1=on	0,1	

	Data type				Custom API variable			Default data type				
	Item	Details	System designation	Axis designation	Section No.	Sub-section No.	Size/ Bit	Data type symbol	Write	Unit	Data range	Remarks
15.7	Stored stroke limit (+) #2014 OT+	The coordinates of the soft limit (–) direction are set. The valid movement range determined by the machine stroke is set. The coordinates must be values that use the basic machine coordinate zero point as the reference point.	O	O	2	288	4	T_LONG	Possible	1=0.5[µm]	–99999.999~ +99999.999 [mm]	
15.8	Stored stroke limit (–) #2013 OT–	The coordinates of the soft limit (+) direction are set.	O	O	2	292	4	T_LONG	Possible	1=0.5[µm]	–99999.999~ +99999.999 [mm]	
15.9	Axis parameter #8202 OT-CHECK OFF	Whether to invalidate the soft limits set with #8204, #8205 is selected. 0: Soft limit valid, 1: Soft limit invalid	O	O	2	640	2/10	T_SHORT	Possible	0=off/1=on	0, 1	
15.10	Axis parameter #8203 OT-CHECK-CANCEL	Whether to cancel the soft limit from after the power is turned ON to the first zero point return when the simple absolute position method is selected is selected. 0: Soft limit valid (Follow #8202) 1: Soft limit temporary cancel	O	O	2	655	1	T_CHAR	Possible	0=off/1=on	0,1	
15.11	Axis parameter #8205 OT-CHECK-P	The coordinates of the soft limit (+) direction are set. (Soft limit is invalid when #8204 = #8205)	O	O	2	656	4	T_LONG	Possible	1=0.5[µm]	–99999.999~ 99999.999 [mm]	
15.12	Axis parameter #8204 OT-CHECK-N	The coordinates of the soft limit (-) direction are set. (Soft limit is invalid when #8204 = #8205)	O	O	2	660	4	T_LONG	Possible	1=0.5[µm]	–99999.999~ 99999.999 [mm]	
15.13	Chuck barrier: X No.1 point #8301	The range of the chuck and tailstock barrier is set. Set with the basic machine coordinate system.	—	—	1	1408($1) 2432($2)	4	T_LONG	Possible	1=0.5[µm]	–99999.999~ 99999.999 [mm]	L/G only
15.14	Chuck barrier: Z No.1 point #8301	The range of the chuck and tailstock barrier is set. Set with the basic machine coordinate system.	—	—	1	1432($1) 2456($2)	4	T_LONG	Possible	1=0.5[µm]	–99999.999~ 99999.999 [mm]	L/G only
15.15	Chuck barrier: X No.2 point #8302	The range of the chuck and tailstock barrier is set. Set with the basic machine coordinate system.	—	—	1	1412($1) 2436($2)	4	T_LONG	Possible	1=0.5[µm]	–99999.999~ 99999.999 [mm]	L/G only
15.16	Chuck barrier: Z No.2 point #8302	The range of the chuck and tailstock barrier is set. Set with the basic machine coordinate system.	—	—	1	1436($1) 2460($2)	4	T_LONG	Possible	1=0.5[µm]	–99999.999~ 99999.999 [mm]	L/G only
15.17	Chuck barrier: X No.3 point #8303	The range of the chuck and tailstock barrier is set. Set with the basic machine coordinate system.	—	—	1	1416($1) 2440($2)	4	T_LONG	Possible	1=0.5[µm]	–99999.999~ 99999.999 [mm]	L/G only

	Data type				Custom API variable			Default data type				
	Item	Details	System designation	Axis designation	Section No.	Sub-section No.	Size/ Bit	Data type symbol	Write	Unit	Data range	Remarks
15.18	Chuck barrier: Z No.3 point #8303	The range of the chuck and tailstock barrier is set. Set with the basic machine coordinate system.	—	—	1	1440($1) 2464($2)	4	T_LONG	Possible	1=0.5[µm]	–99999.999~ 99999.999 [mm]	L/G only
15.19	Chuck barrier: X No.4 point #8304	The range of the chuck and tailstock barrier is set. Set with the basic machine coordinate system.	—	—	1	1420($1) 2444($2)	4	T_LONG	Possible	1=0.5[µm]	–99999.999~ 99999.999 [mm]	L/G only
15.20	Chuck barrier: Z No.4 point #8304	The range of the chuck and tailstock barrier is set. Set with the basic machine coordinate system.	—	—	1	1444($1) 2468($2)	4	T_LONG	Possible	1=0.5[µm]	–99999.999~ 99999.999 [mm]	L/G only
15.21	Chuck barrier: X No.5 point #8305	The range of the chuck and tailstock barrier is set. Set with the basic machine coordinate system.	—	—	1	1424($1) 2448($2)	4	T_LONG	Possible	1=0.5[µm]	–99999.999~ 99999.999 [mm]	L/G only
15.22	Chuck barrier: Z No.5 point #8305	The range of the chuck and tailstock barrier is set. Set with the basic machine coordinate system.	—	—	1	1448($1) 2472($2)	4	T_LONG	Possible	1=0.5[µm]	–99999.999~ 99999.999 [mm]	L/G only
15.23	Chuck barrier: Z No.6 point #8306	The range of the chuck and tailstock barrier is set. Set with the basic machine coordinate system.	—	—	1	1428($1) 2452($2)	4	T_LONG	Possible	1=0.5[µm]	–99999.999~ 99999.999 [mm]	L/G only
15.24	Chuck barrier: X No.6 point #8306	The range of the chuck and tailstock barrier is set. Set with the basic machine coordinate system.	—	—	1	1452($1) 2476($2)	4	T_LONG	Possible	1=0.5[µm]	–99999.999~ 99999.999 [mm]	L/G only
15.25	Axis name of position switch [n]: $1	The name of the axis for which a hypothetical dog position (position switch) is set.	—	—	1	11964 +(20*n) (n=1~8)	1	T_CHAR	Possible		X,Y,Z,U,V,W, A,B,C	
15.26	Dog position (1st axis) of position switch [n]: $1	The coordinate position used as the hypothetical dog position is set.	—	—	1	11948 +(20*n) (n=1~8)	4	T_LONG	Possible	1=0.5[µm]	–99999.999~ 99999.999 [mm]	
15.27	Dog position (2nd axis) of position switch [n]: $1	The coordinate position used as the hypothetical dog position is set.	—	—	1	11952 +(20*n) (n=1~8)	4	T_LONG	Possible	1=0.5[µm]	–99999.999~ 99999.999 [mm]	
15.28	Axis name of position switch [n]: $2	The name of the axis for which a hypothetical dog position (position switch) is set.	—	—	1	12124 +(20*n) (n=1~8)	1	T_CHAR	Possible		X,Y,Z,U,V,W, A,B,C	
15.29	Dog position (1st axis) of position switch [n]: $2	The coordinate position used as the hypothetical dog position is set.	—	—	1	12108 +(20*n) (n=1~8)	4	T_LONG	Possible	1=0.5[µm]	–99999.999~ 99999.999 [mm]	
15.30	Dog position (2nd axis) of position switch [n]: $2	The coordinate position used as the hypothetical dog position is set.	—	—	1	12112 +(20*n) (n=1~8)	4	T_LONG	Possible	1=0.5[µm]	–99999.999~ 99999.999 [mm]	

	Data type				Custom API variable			Default data type				
	Item	Details	System designation	Axis designation	Section No.	Sub-section No.	Size/ Bit	Data type symbol	Write	Unit	Data range	Remarks
15.31	Rotation axis #1017 rot	Whether the axis is a rotation axis (1) or linear axis (0) is designated. If the rotation axis is selected, the position display will return to 0 deg. after 360 deg. Designate the linear axis when the position is to be displayed continuously.	O	O	2	20	2/4	T_SHORT	Possible		0,1	
15.32	Motor CCW control axis. #1018 ccw	The motor rotation direction to the commanded direction is designated. 0: Rotate in clockwise direction (looking from motor shaft) with forward direction command. 1: Rotate in counter clockwise direction (looking from motor shaft) with reverse direction command.	O	O	2	20	2/5	T_SHORT	Possible		0,1	
15.33	Axis exchangeable axis	Whether the axis can be exchanged is designated. 1: Exchangeable axis 0: Non-exchangeable axis	O	O	2	20	2/7	T_SHORT	Possible		0,1	
15.34	Diameter designation axis #1019 dia	The axis handling method is designated as diameter or radius. 1: Diameter designation, 0: Radius designation	O	O	2	20	2/13	T_SHORT	Possible		0,1	L/G only
15.35	Reference point return approach speed #2026 G28crp	Set the approach speed used after the dog is detected during dog-type reference point return.	O	O	2	38	2	T_SHORT	Possible	1=1[mm/min]	1~60000 [mm/min]	
15.36	Reference point return shift amount #2027 G28sft	Set the shift amount when the reference point determined by the detector reference point is to be shifted during reference point return.	O	O	2	44	2	T_SHORT	Possible	1=1[µm]	0~65535 [µm]	
15.37	Rapid traverse backlash amount #2011 G0back	Set the backlash offset amount during rapid traverse. The setting range is –9999 to 9999 (0.5µm units)	O	O	2	46	2	T_SHORT	Possible	1=0.5[µm]	–9999 ~+9999	
15.38	Cutting backlash amount #2012 G1back	Set the backlash offset amount during cutting feed. The setting range is –9999 to 9999 (0.5µm units)	O	O	2	48	2	T_SHORT	Possible	1=0.5[µm]	–9999 ~+9999	
15.39	Compensation basic axis #4001 cmpax #4011...	Basic axis address for machine error compensation (During pitch error compensation: Name of axis to be compensated) (During incremental position error compensation: Name of axis to be used as reference)	—	—	3	0+32*n (n=0~4)	2	T_SHORT	Possible		X,Y,Z,U,V,W, A,B,C	Space + axis name(x) = 2058 (Hex) = 8280 (Dec)

	Data type				Custom API variable			Default data type				
	Item	Details	System designation	Axis designation	Section No.	Sub-section No.	Size/ Bit	Data type symbol	Write	Unit	Data range	Remarks
15.40	Compensation direction axis #4002 drcax #4012...	The compensation axis address for machine error compensation is designated. (During pitch error compensation: Same axis name as #4001) (During incremental position error compensation: Name of axis to be used as reference for incremental position compensation)	—	—	3	4+32*n (n=0~4)	2	T_SHORT	Possible		X,Y,Z,U,V,W, A,B,C	Space + axis name(x) = 2058 (Hex) = 8280 (Dec)
15.41	Reference point division point No. #4003 rdvno #4013...	No. of compensation data corresponding to reference point position	—	—	3	8+32*n (n=0~4)	2	T_SHORT	Possible		4101~5124	
15.42	Far minus-side division point No. #4004 mdvno #4014...	Minus side compensation data No.	—	—	3	12+32*n (n=0~4)	2	T_SHORT	Possible		4101~5124	
15.43	Far plus-side division point No. #4005 pdvno #4015...	Plus side compensation data No.	—	—	3	16+32*n (n=0~4)	2	T_SHORT	Possible		4101~5124	
15.44	Compensation scale #4006 sc #4016...	Scale of the compensation amount	—	—	3	20+32*n (n=0~4)	2	T_SHORT	Possible		0~99	
15.45	Division interval #4007 spcdy #4017...	Interval to divide reference axis (Each compensation data is the compensation amount per interval)	—	—	3	28+32*n (n=0~4)	4	T_LONG	Possible	1=0.5[µm]	1~9999999	
15.46	Machine error compensation parameter #4101~#5124	Machine error compensation amount	—	—	3	384+n–4101 (n=4101 ~5123)	2	T_SHORT	Possible			n is an add number (4101, 4103, ...) Refer to Chapter 4.2
15.47	No. of workpiece machining times control #8001 WRK COUNT M	The M code for counting the No. of times the workpiece is repeatedly machined is set. The setting range is 6~29, 31~97. (The times will not be counted when set to 0.)	—	—	1	1280($1) 2304($2)	1	T_CHAR	Possible	1=1[set]	6~29,31~97	This is an M code not used as a miscellaneous function or macro function.
15.48	No. of workpiece machining times control #8002 WRK COUNT	The initial value setting for the No. of machining times/current No. of machining times is retrieved.	—	—	10318	2896($1) 1892($2)	4	T_LONG	Possible	1=1[set]	0~999999	
15.49	No. of workpiece machining times control #8003 WRK LIMIT	The max. value of workpiece machining times is set. (When No. of machining times = Max. No. of times, PLC signal will be output.)	—	—	10318	2898($1) 1894($2)	4	T_LONG	Possible	1=1[set]	0~999999	

	Data type				Custom API variable			Default data type				
	Item	Details	System designation	Axis designation	Section No.	Sub-section No.	Size/ Bit	Data type symbol	Write	Unit	Data range	Remarks
16. Data related to NC serial input/output												
16.1	RS-232-C Port No. #9001 for data input	The port No. for inputting part programs and parameters, etc., is designated. 1: SIO1, 2: Using SIO2 connector	—	—	1	12288	1	T_CHAR	Possible		1,2	
16.2	RS-232-C Device No. #9002 for data input	The device No. for inputting part programs and parameters, etc., is designated. The setting range is 0 to 4.	—	—	1	12304	1	T_CHAR	Possible		0~4	
16.3	RS-232-C Port No. #9003 for data output	The port No. for outputting part programs and parameters, etc., is designated. 1: SIO1, 2: Using SIO2 connector	—	—	1	12289	1	T_CHAR	Possible		1,2	
16.4	RS-232-C Device No. #9004 for data output	The device No. for outputting part programs and parameters, etc., is designated. The setting range is 0 to 4.	—	—	1	12305	1	T_CHAR	Possible		0~4	
16.5	RS-232-C Port No. #9005 for tape operation	The port No. for input during tape operation is designated. 1: SIO1, 2: Using SIO2 connector	—	—	1	12290	1	T_CHAR	Possible		1,2	
16.6	RS-232-C Device No. #9006 for tape operation	The device No. for input during tape operation is designated. The setting range is 0 to 4.	—	—	1	12306	1	T_CHAR	Possible		0~4	
16.7	RS-232-C Port No. #9007 for macro print	The port No. for outputting with user macro DPRINT command is designated. 1: SIO1, 2: Using SIO2 connector	—	—	1	12291	1	T_CHAR	Possible		1,2	
16.8	RS-232-C Device No. #9008 for macro print	The device No. for outputting with user macro DPRINT command is designated. The setting range is 0 to 4.	—	—	1	12307	1	T_CHAR	Possible		0~4	
16.9	RS-232-C Port No. #9009 for PLC input/output	The port No. for inputting/outputting various data with the PLC is designated. 1: SIO1, 2: Using SIO2 connector	—	—	1	12292	1	T_CHAR	Possible		1,2	
16.10	RS-232-C Device No. #9010 for PLC input/ output	The device No. for inputting/outputting various data with the PLC is designated. The setting range is 0 to 4.	—	—	1	12308	1	T_CHAR	Possible		0~4	
16.11	RS-232-C Input/output device environment setting #9101 DEVICE NAME	The device name is designated for device No. n. The setting range is within 3 alphanumeric characters.	—	—	1	12320 +(48∗n) (n = 0~4)	3	T_STR	Possible			Device name 3 letters + NULL
16.12	RS-232-C Input/output device environment setting #9102 BAUD RATE	The serial communication baud rate is designated for device No. n. 1:9600bps 2:4800bps 3:2400bps 4:1200bps 5:600bps 6:300bps 7:150bps	—	—	1	12324 +(48∗n) (n = 0~4)	1	T_CHAR	Possible		1~7	

	Data type				Custom API variable			Default data type				
	Item	Details	System designation	Axis designation	Section No.	Sub-section No.	Size/ Bit	Data type symbol	Write	Unit	Data range	Remarks
16.13	RS-232-C Input/output device environment setting #9103 STOP BIT	The stop bit length during start-stop communication corresponding is designated for device No. n. 1: 1 bit 2: 1.5 bit 3: 2 bit	—	—	1	12325 +(48*n) (n = 0~4)	1	T_CHAR	Possible		1~3	
16.14	RS-232-C Input/output device environment setting #9104 PARITY CHECK	The addition of a parity check bit to data during communication is designated for device No. n. 0: No parity bit, 1: With parity bit	—	—	1	12327 +(48*n) (n = 0~4)	1	T_CHAR	Possible		0,1	
16.15	RS-232-C Input/output device environment setting #9105 EVEN PARITY	The odd/even parity when a parity check bit is added is designated for device No. n. 0: Odd parity, 1: Even parity	—	—	1	12326 +(48*n) (n = 0~4)	1	T_CHAR	Possible		0,1	
16.16	RS-232-C Input/output device environment setting #9106 CHR. LENGTH	The data bit length is set for device No. n. 0: Data is 5-bit 1: Data is 6-bit 2: Data is 7-bit 3: Data is 8-bit	—	—	1	12328 +(48*n) (n = 0~4)	1	T_CHAR	Possible		0~3	
16.17	RS-232-C Input/output device environment setting #9108 HAND SHAKE	The conveyance control method is designated for device No. n. 1: RTS/CTS method (Can be used only for SIO2) 2: No order (No handshake) 3: DC code method	—	—	1	12336 +(48*n) (n = 0~4)	1	T_CHAR	Possible		1~3	
16.18	RS-232-C Input/output device environment setting #9109 DC CODE PARITY	The DC parity bit addition when DC code method is selected is set for device No. n. 0: No parity for DC code (DC3 = 13H) 1: Parity for DC code (DC3 = 93H)	—	—	1	12337 +(48*n) (n = 0~4)	1	T_CHAR	Possible		0,1	
16.19	RS-232-C Input/output device environment setting #9111 DC2/DC4 OUTPUT	The handling of the DC code during output of data to device No. n output device is designated. DC2/DC4 0: No/No 1: Yes/No 2: No/Yes 3: Yes/Yes	—	—	1	12347 +(48*n) (n = 0~4)	1	T_CHAR	Possible		0~3	
16.20	RS-232-C Input/output device environment setting #9112 CR OUTPUT	Addition of a <CR> code before the EOB (L/F) code during output to device No. n is designated. 0: Not added, 1: Added	—	—	1	12334 +(48*n) (n = 0~4)	1	T_CHAR	Possible		0,1	
16.21	RS-232-C Input/output device environment setting #9114 FEED CHR.	The length of the tape freed output at the head and end of the tape during tape output to the device No. n is designated. The setting range is 0 to 999 characters.	—	—	1	12348 +(48*n) (n = 0~4)	2	T_SHORT	Possible		0~999	
16.22	RS-232-C Input/output device environment setting #9115 PARITY V	The execution of a parity check for the number of characters in one block during tape input from the device No. n is designated. 0: No parity V check, 1: With parity V check	—	—	1	12352 +(48*n) (n = 0~4)	1/1	T_CHAR	Possible		0,1	
16.23	RS-232-C Input/output device environment setting #9116 TIME-OUT (sec.)	The timeout time for detection an interruption in the communication is designated for device No. n. The setting range is 0 to 30 (sec.) (Timeout will not be executed when 0 is set.)	—	—	1	12350 +(48*n) (n = 0~4)	2	T_SHORT	Possible		0~30	

	Data type				Custom API variable			Default data type				
	Item	Details	System designation	Axis designation	Section No.	Sub-section No.	Size/ Bit	Data type symbol	Write	Unit	Data range	Remarks
17. Data related to NC specification tuning												
17.1	Valid system	This is used to check whether the system is valid or not. 1: Valid system, 0: Invalid system	—	—	1	512($1) 1536($2)	1	T_CHAR	Not possible		0,1	
17.2	Inch unit #1041 I_inch	The default of the NC machining program unit system is set to inch (G20). 1: Valid, 0: Invalid	O	—	1	192	1/4	T_CHAR	Possible	0=off/1=on	0,1	
17.3	Input unit #1003	The input unit of the system axis is set. A : 10µm, B : 1µm, C : 0.1µm	—	—	1	516($1) 1540($2)	1	T_CHAR	Not possible		A,B,C	0.1µm is an option.
17.4	Synchronous tap	Validity of synchronous tap command. 0: Invalid, 1: Valid	—	—	1	216	1/4	T_CHAR	Possible	0=off/1=on	0,1	
17.5	Servo OFF error correction #1064 svof	Whether to correct the error during servo OFF is designated. 1: Do not correct error, 1: Correct error	O	O	2	20	2/3	T_SHORT	Possible	0=off/1=on	0,1	
17.6	Origin set invalid	The origin set function validity is changed over. 1: Invalid, 0: Valid (normal)	—	—	1	130	1/2	T_CHAR	Possible	0=off/1=on	0,1	
17.7	Reference point return: Middle point ignore #1091 Mpoint	Designation of transit point during G28, G30 reference point return. 0: Program designated point, 1: No transit point	—	—	1	194	1/5	T_CHAR	Possible	0=off/1=on	0,1	
17.8	Control parameter #8101 MACRO SINGLE	The control of the block where the user macro command continues is selected. 0: Do not stop in area where macro block continues. 1: Stop at each block during single blocks.	—	—	1	48101	4	T_LONG	Possible	0=off/1=on	0,1	Usable because of the API function (1-194) compatibility, but do not use.
17.9	Control parameter #8102 COLL. ALM OFF	The interference control to the workpiece according to the tool diameter for tool diameter and nose R offset is set. 0: Stop with an alarm when judged that interference will occur. 1: Change path so that interference does not occur.	—	—	1	48102	4	T_LONG	Possible	0=off/1=on	0,1	Usable because of the API function (1-193) compatibility, but do not use.
17.10	Control parameter #8103 COLL. CHK OFF	The interference control to the workpiece according to the tool diameter for tool diameter and nose R offset is set. 0: Perform interference check 1: Do not perform interference check	—	—	1	48103	4	T_LONG	Possible	0=off/1=on	0,1	Usable because of the API function (1-198) compatibility, but do not use.
17.11	Control parameter #8106 G46 NO REV-ERR	The control for the compensation direction reversal with G46 (nose R offset) is set. 0: Stop with alarm when compensation direction is reversed (G41→G42/G42→G41) 1: Do not generate alarm when compensation direction is reversed, and operate in that compensation direction.	—	—	1	48106	4	T_LONG	Possible	0=off/1=on	0,1	Usable because of the API function (1-196) compatibility, but do not use.

	Data type				Custom API variable			Default data type				
	Item	Details	System designation	Axis designation	Section No.	Sub-section No.	Size/ Bit	Data type symbol	Write	Unit	Data range	Remarks
17.12	Control parameter #8107 Radius error compensation	Selection of the arc radius error compensation control 0: Compensation OFF (During circular cutting, moves to the inside by the servo delay, etc., in respect to the command, and the arc becomes smaller than the command value) 1: Compensation ON (During circular cutting, compensates the movement to the inside by the servo delay.)	—	—	1	48107	4	T_LONG	Possible	0=off/1=on	0,1	
17.13	#1100 Tmove	The period to execute tool length offset and wear offset operation is set. [0] Execute compensation during T command execution. [1] Execute compensation in superimposition with movement command in same block as T command. Next movement command is superimposed when there is no movement in the same block.	—	—	1	100	1/2	T_CHAR	Possible	0=off/1=on	0,1	
17.14	#1101 Tabsmv	The type of movement command when #1100=1 is designated. [0] Compensate with all movement commands [1] Compensate only with absolute value command movement commands	—	—	1	130	1/1	T_CHAR	Possible	0=off/1=on	0,1	
17.15	#1099 Treset	The handling of the tool offset vector during reset is designated. [0] Clear tool length and wear offset vector with reset [1] Hold tool length and wear offset vector with reset	—	—	1	100	1/1	T_CHAR	Possible	0=off/1=on	0,1	
17.16	Automatic tool length measurement #8004 SPEED	The feedrate during automatic tool measurement is set.	—	—	1	1244($1) 2268($2)	4	T_LONG	Possible	1=1[mm/min]	1~60000 [mm/min]	
17.17	Automatic tool length measurement #8005 ZONE r	The distance between the measurement point and deceleration start point is set.	—	—	1	1236($1) 2260($2)	4	T_LONG	Possible	1=0.5[µm]	0~ 99999.999 [mm]	
17.18	Automatic tool length measurement #8006 ZONE d	The tolerable range of the measurement point is set. An alarm will occur if the sensor signal turns ON d or more before the measurement point, or if the sensor signal does not turn ON even when d is passed.	—	—	1	1240($1) 2264($2)	4	T_LONG	Possible	1=0.5[µm]		
17.19	Automatic corner override #8007 OVERRIDE	The override value for automatic corner override is set.	—	—	1	1282($1) 2306($2)	1	T_CHAR	Possible		0~100[%]	
17.20	Automatic corner override #8008 MAX ANGLE	The max. opening angle of the corner where automatic deceleration is to be applied is set.	—	—	1	1156($1) 2180($2)	4	T_LONG	Possible	1=1[deg]	0~180[deg]	
17.21	Automatic corner override #8009 DSC. ZONE	The deceleration start position at the corner (distance to decelerate before corner) is set.	—	—	1	1160($1) 2184($2)	4	T_LONG	Possible	1=0.5[µm]	0~99999.999 [mm]	

	Data type				Custom API variable			Default data type				
	Item	Details	System designation	Axis designation	Section No.	Sub-section No.	Size/ Bit	Data type symbol	Write	Unit	Data range	Remarks
17.22	Fixed cycle #8012 G73 n	The return amount for G73 (step cycle) is set.	—	—	1	1216($1) 2240($2)	4	T_LONG	Possible	1=1[µm]	0~99999.999 [mm]	
17.23	Fixed cycle #8013 G83 n	The return amount for G83 (deep hole drill) is set.	—	—	1	1232($1) 2256($2)	4	T_LONG	Possible	1=1[µm]	0~99999.999 [mm]	
17.24	Fixed cycle #8014 CDZ-VALE	The thread cutting up amount at G76, G78 (thread cutting cycle) is set.	—	—	1	1284($1) 2308($2)	1	T_CHAR	Possible	1=0.1[Read]	0~12.7 [Read]	L only
17.25	Fixed cycle #8015 CDZ-ANGLE	The thread cutting up degree at G76, G78 (thread cutting cycle) is set.	—	—	1	1283($1) 2307($2)	1	T_CHAR	Possible	1=1[deg]	0~89[deg]	L only
17.26	Fixed cycle #8016 G71 MINIMUM	The min. value of the final cutting up amount at G71, G72 (rough cutting cycle) is set.	—	—	1	1188($1) 2212($2)	4	T_LONG	Possible	1=0.5[µm]	0~99.999 [mm]	L/G only
17.27	Fixed cycle #8017 DELTA-D	The change amount to the commanded cut amount D at G71, G72 (rough cutting cycle) is set.	—	—	1	1192($1) 2216($2)	4	T_LONG	Possible	1=0.5[µm]	0~99.999 [mm]	L/G only
17.28	Fixed cycle #8018 G84/G74 return	Becomes a normal tap cycle by setting the return amount of the G84/G74 pecking tap cycle to "0".	—	—	1	1212($1) 2236($2)	4	T_LONG	Possible	1=1[µm]	0~999.999 [mm]	M/D only
17.29	High precision # 8019 Number of precision systems	Number of compensation systems when an even smaller control error of the corner radius and arc radius reduction, etc., is required. Theoretically, the precision error becomes smaller as the setting value gets larger, but the cycle time is longer because of the lower speed at the corners.	O	—	1	48019	4	T_LONG	Possible	1=1[%]	0~99[%]	
17.30	High precision #8020 Corner deceleration angle	Minimum value of an angle (external angle) to be recognized as a corner.	O	—	1	48020	4	T_LONG	Possible	1=1[deg]	0~10[deg] 0: Automatic determination	
17.31	Axis parameter #8201 AX. RELEASE	Whether the control axis is to be removed from the control target is selected. 0: Normal, 1: Remove from control target	O	O	2	640	2/6	T_CHAR	Possible	0=off/1=on	0,1	
17.32	Axis parameter #8206 TOOL CHG.P	The tool exchange position coordinates at G30.n are set. (Set with basic machine coordinate system.)	O	O	2	668	4	T_LONG	Possible	1=0.5[µm]	–99999.999~ 99999.999 [mm]	
17.33	Axis parameter #8207 G76/87 IGNR	The shift operation during G76 (fine boring) or G87 (back boring) is selected. 0: No shift operation, 1: Shift operation	O	O	2	640	2/11	T_SHORT	Possible	0=off/1=on	0,1	M/D only
17.34	Axis parameter #8208 G76/87 (–)	The shift direction during G76, G87 is designated. 0: Shift in + direction, 1: Shift in – direction	O	O	2	640	2/12	T_SHORT	Possible	0=off/1=on	0,1	M/D only
17.35	Axis parameter #8209 G60 SHIFT	Final positioning direction and distance during G60 (uni-direction positioning).	O	O	2	752	4	T_LONG	Possible	1=0.5[µm]	–99999.999~ 99999.999 [mm]	M/D only
17.36												

	Data type				Custom API variable			Default data type				
	Item	Details	System designation	Axis designation	Section No.	Sub-section No.	Size/ Bit	Data type symbol	Write	Unit	Data range	Remarks
17.37	Second miscellaneous function name #1170 M2name	The address used for the second miscellaneous function command (Use A, B or C and make sure it is not duplicated with the axis name.)	—	—	1	569($1) 1593($2)	1	T_CHAR	Possible		A, B or C	
17.38	Machining program registration capacity	The max. number of machining programs that can be registered is retrieved.	—	—	1	32	1	T_CHAR	Not possible	1=64[Program] 2=128[Program]	64,128 [Program]	The data range depends on the NC system specifications.
17.39	Number of registered machining programs	The number of programs registered as user machining programs is displayed.	—	—	1	40002	4	T_LONG	Not possible	1=1[Program]		
17.40	Number of remaining machining programs	The number of machining programs that can be registered is retrieved.	—	—	1	40003	4	T_LONG	Not possible	1=1[Program]		
17.41	Number of registered characters	The number of characters registered as user machining programs is displayed.	—	—	1	40004	4	T_LONG	Not possible	1=1 [Character]		
17.42	Machining program remaining capacity	The remaining number of characters that can be registered is displayed. (Displayed in 250-character units.)	—	—	1	40005	4	T_LONG	Not possible	1=1 [Character]		
17.43	Measuring tool length	Manual tool length measurement 1 is validated, or it indicates that manual tool length measurement 1 is valid.	O	—	10025($1) 10023($2)	552($1) 168($2)	/0	T_BIT	Not possible			
17.44	In automatic operation	Indicates that operation is being executed with automatic operation start up. This is used when parameters are set during automatic operation, the tool offset is changed, the common variables are changed, or the workpiece offset changes are locked.	O	—	10024($1) 10021($2)	498($1) 114($2)	/0	T_BIT	Not possible			
17.45	In automatic operation start up	Indicates that the program has been started with automatic operation start up, and that a movement command, M, S or T command or dwell command is being executed.	O	—	10024($1) 10021($2)	499($1) 115($2)	/0	T_BIT	Not possible			
18. Data related to NC system												
18.1	NC system No.	NC system No. and control software version. BND-353W000-A0	—	—	1	40001	16	T_STR	Not possible			
18.2	Number of systems	Method for calculating number of systems When [1=1536] is 0, 1 system, and when 1, 2 systems.	—	—	1	1536	1	T_CHAR	Not possible	0=1[system]	0,1	
18.3	NC system name	MELDASMAGIC 50M	—	—	1	40010	20	T_STR	Not possible			

	Data type				Custom API variable			Default data type				
	Item	Details	System designation	Axis designation	Section No.	Sub-section No.	Size/ Bit	Data type symbol	Write	Unit	Data range	Remarks
19. PLC device												
19.1	Modal write release	PLC device modal write release	—	—	20000	0	2	T_SHORT				Read not possible Refer to Chapter 4.3
19.2	Deive B counter (Fixed counter)	Device B read/write in 1-bit units	—	—	10002	0~103	/0	T_BIT				
19.3		Device B read/write in 8-bit units	—	—	10102	0~96	1	T_CHAR				
19.4		Device B read/write in 16-bit units	—	—	10202	0~80	2	T_SHORT				
19.5		Device B read/write in 32-bit units	—	—	10302	0~64	4	T_LONG				
19.6		Device B modal write in 1-bit units	—	—	11002	0~103	/0	T_BIT				
19.7	Device C Counter	Device C read/write in 1-bit units	—	—	10003	0~23	/0	T_BIT				
19.8		Device C read/write in 8-bit units	—	—	10103	0, 8, 16	1	T_CHAR				
19.9		Device C read/write in 16-bit units	—	—	10203	0	2	T_SHORT				
19.10		Device C modal write in 1-bit units	—	—	11003	0~23	/0	T_BIT				
19.11	Device D Data register	Device D read/write in 2-byte units	—	—	10204	0~1023	2	T_SHORT				
19.12		Device D read/write in 4-byte units	—	—	10304	0~1022	4	T_LONG				
19.13		Device D modal write in 2-byte units	—	—	11204	0~1023	2	T_SHORT				
19.14	Defice E Special relay	Device E read/write in 1-bit units	—	—	10005	0~127	/0	T_BIT				
19.15		Device E read/write in 8-bit units	—	—	10105	0~120	1	T_CHAR				
19.16		Device E read/write in 16-bit units	—	—	10205	0~12	2	T_SHORT				
19.17		Device E read/write in 32-bit units	—	—	10305	0~96	4	T_LONG				
19.18		Device E modal write in 1-bit units	—	—	11005	0~127	/0	T_BIT				
19.19	Device F Alarm message Interface Temporary memory	Device F read/write in 1-bit units	—	—	10006	0~127	/0	T_BIT				
19.20		Device F read/write in 8-bit units	—	—	10106	0~120	1	T_CHAR				
19.21		Device F read/write in 16-bit units	—	—	10206	0~112	2	T_SHORT				
19.22		Device F read/write in 32-bit units	—	—	10306	0~96	4	T_LONG				
19.23		Device F modal write in 1-bit units	—	—	11006	0~127	/0	T_BIT				
19.24	Device G Temporary memory	Device G read/write in 1-bit units	—	—	10007	0~3071	/0	T_BIT				
19.25		Device G read/write in 8-bit units	—	—	10107	0~3064	1	T_CHAR				
19.26		Device G read/write in 16-bit units	—	—	10207	0~3056	2	T_SHORT				
19.27		Device G read/write in 32-bit units	—	—	10307	0~3040	4	T_LONG				
19.28		Device G modal write in 1-bit units	—	—	11007	0~3071	/0	T_BIT				

	Data type				Custom API variable			Default data type				
	Item	Details	System designation	Axis designation	Section No.	Sub-section No.	Size/ Bit	Data type symbol	Write	Unit	Data range	Remarks
19.29	Device I	Device I read/write in 1-bit units	—	—	10009	0~1215	/0	T_BIT				
19.30		Device I read/write in 1-bit units	—	—	10109	0~1208	1	T_CHAR				
19.31		Device I read/write in 1-bit units	—	—	10209	0~1200	2	T_SHORT				
19.32		Device I read/write in 1-bit units	—	—	10309	0~1184	4	T_LONG				
19.33		Device I modal write in 1-bit units	—	—	11009	0~1215	/0	T_BIT				
19.34	Device J	Device J read/write in 1-bit units	—	—	10010	0~1599	/0	T_BIT				
19.35		Device J read/write in 1-bit units	—	—	10110	0~1592	1	T_CHAR				
19.36		Device J read/write in 1-bit units	—	—	10210	0~1584	2	T_SHORT				
19.37		Device J read/write in 1-bit units	—	—	10310	0~1568	4	T_LONG				
19.38		Device J modal write in 1-bit units	—	—	11010	0~1599	/0	T_BIT				
19.39	Device L Latch relay Back up memory	Device L read/write in 1-bit units	—	—	10012	0~255	/0	T_BIT				
19.40		Device L read/write in 1-bit units	—	—	10112	0~248	1	T_CHAR				
19.41		Device L read/write in 1-bit units	—	—	10212	0~240	2	T_SHORT				
19.42		Device L read/write in 1-bit units	—	—	10312	0~224	4	T_LONG				
19.43		Device L modal write in 1-bit units	—	—	11012	0~255	/0	T_BIT				
19.44	Device M Temporary memory	Device M read/write in 1-bit units	—	—	10013	0~5119	/0	T_BIT				
19.45		Device M read/write in 1-bit units	—	—	10113	0~5112	1	T_CHAR				
19.46		Device M read/write in 1-bit units	—	—	10213	0~5104	2	T_SHORT				
19.47		Device M read/write in 1-bit units	—	—	10313	0~5088	4	T_LONG				
19.48		Device M modal write in 1-bit units	—	—	11013	0~5119	/0	T_BIT				
19.49	Device Q	10msec unit timer	—	—	10017	0~39	/0	T_BIT				
19.50		100msec unit timer	—	—		40~1356	/0	T_BIT				
19.51		100msec increment timer	—	—		136~151	/0	T_BIT				
19.52		10msec unit timer (read/write in 8-bit units)	—	—	10117	0~32	1	T_CHAR				
19.53		100msec unit timer (read/write in 8-bit units)	—	—		40~128	1	T_CHAR				
19.54		100msec increment timer (read/write in 8-bit units)	—	—		136~144	1	T_CHAR				
19.55		10msec unit timer (read/write in 16-bit units) Q0-Q15, Q16-Q31	—	—	10217	0, 16	2	T_SHORT				
19.56		10msec unit timer modal write	—	—	11017	0~39	/0	T_BIT				
19.57		100msec unit timer modal write	—	—		40~135	/0	T_BIT				
19.58		100msec increment timer modal write	—	—		136~151	/0	T_BIT				

	Data type				Custom API variable			Default data type				
	Item	Details	System designation	Axis designation	Section No.	Sub-section No.	Size/ Bit	Data type symbol	Write	Unit	Data range	Remarks
19.59	Device R File Register	Device R read/write in 2-byte units	—	—	10218	0~8191	2	T_SHORT				
19.60		Device R read/write in 4-byte units	—	—	10318	0~8190	4	T_LONG				
19.61		Device R modal write in 2-byte units	—	—	10218	0~8191	2	T_SHORT				
19.62	Device S	Device S read/write in 1-byte units	—	—	10019	0~511	/0	T_BIT				
19.63		Device S read/write in 8-byte units	—	—	10119	0~504	1	T_CHAR				
19.64		Device S read/write in 16-byte units	—	—	10219	0~496	2	T_SHORT				
19.65		Device S read/write in 32-byte units	—	—	10319	0~480	4	T_LONG				
19.66		Device S modal write in 1-bit units	—	—	11019	0~511	/0	T_BIT				
19.67	Device T Timer	10msec unit timer	—	—	10020	0~15	/0	T_BIT				
19.68		100msec unit timer	—	—		16~95	/0	T_BIT				
19.69		100msec increment timer	—	—		96~103	/0	T_BIT				
19.70		10msec unit timer (read/write in 8-bit units)	—	—	10120	0, 8	1	T_CHAR				
19.71		100msec unit timer (read/write in 8-bit units)	—	—		16~88	1	T_CHAR				
19.72		100msec increment timer (read/write in 8-bit units)	—	—		96	1	T_CHAR				
19.73		10msec unit timer (read/write in 16-bit units) T0-T15	—	—	10220	0	2	T_SHORT				
19.74		10msec unit timer (read/write in 16-bit units) T16-T31, T32-T47, T48-T63, T64-T79, T80-T95	—	—		16~80	2	T_SHORT				
19.75		10msec unit timer modal write	—	—	11020	0~15	/0	T_BIT				
19.76		100msec unit timer modal write	—	—		16~95	/0	T_BIT				
19.77		100msec increment timer modal write	—	—		96~103	/0	T_BIT				
19.78	Device U Input signal to PLC For 2-system	Device U read/write in 1-bit units	—	—	10021	0~383	/0	T_BIT				
19.79		Device U read/write in 8-bit units	—	—	10121	0~376	1	T_CHAR				
19.80		Device U read/write in 16-bit units	—	—	10221	0~368	2	T_SHORT				
19.81		Device U read/write in 32-bit units	—	—	10321	0~352	4	T_LONG				
19.82		Device U modal write in 1-bit units	—	—	11021	0~383	/0	T_BIT				
19.83	Device W Output signal to PLC For 2-system	Device W read/write in 1-bit units	—	—	10023	0~511	/0	T_BIT				
19.84		Device W read/write in 8-bit units	—	—	10123	0~504	1	T_CHAR				
19.85		Device W read/write in 16-bit units	—	—	10223	0~496	2	T_SHORT				
19.86		Device W read/write in 32-bit units	—	—	10323	0~480	4	T_LONG				
19.87		Device W read/write in 1-bit units	—	—	11023	0~511	/0	T_BIT				

	Data type				Custom API variable			Default data type				
	Item	Details	System designation	Axis designation	Section No.	Sub-section No.	Size/ Bit	Data type symbol	Write	Unit	Data range	Remarks
19.88		Device W read/write in 1-bit units	—	—	10024	0~1279	/0	T_BIT				
19.89	Device X	Device W read/write in 8-bit units	—	—	10124	0~1272	1	T_CHAR				
19.90	Input signal to PLC	Device W read/write in 16-bit units	—	—	10224	0~1264	2	T_SHORT				
19.91	For 1-system	Device W read/write in 32-bit units	—	—	10324	0~1248	4	T_LONG				
19.92		Device X modal write in 1-bit units	—	—	11024	0~1279	/0	T_BIT				
19.93		Device Y read/write in 1-bit units	—	—	10025	0~1407	/0	T_BIT				
19.94	Device Y	Device Y read/write in 8-bit units	—	—	10125	0~1400	1	T_CHAR				
19.95	Output signal to PLC	Device Y read/write in 16-bit units	—	—	10225	0~1392	2	T_SHORT				
19.96	For 1-system	Device Y read/write in 32-bit units	—	—	10325	0~1376	4	T_LONG				
19.97		Device X modal write in 1-bit units	—	—	11025	0~1407	/0	T_BIT				

4. Detailed Explanation of Individual Functions

4.1 Common variable and variable <Open> (#0)

A variable <Open> state exists when absolutely no value is set in a common variable (#100~, #500~) or local variable (#1~#33).
The machining program interpretation differs for when the variable is 0. and when the variable is <Open>. The different interpretations are shown below.

For example, when #100 = 0. , program 1 and program 2 have the same meaning.

```
(Program 1)
  G01 X#100 F1000;
```

```
(Program 2)
  G01 X0. F1000;
```

However, when #100 = 0 (<Open>), program 1 has the same meaning as program 3.

```
(Program 3)
  G01 F1000;
```

In other words, when the variable is 0. it is interpreted as a zero command, but when the variable is in an <Open> status it is interpreted as not having the address X command.

When <Open> is set in the common variable or local variable **(Note 1)** as shown above, the value read by melReadData and melReadModal cannot be operated or displayed. This is because the variable <Open> uses the IEEE standard "Not a number". When reading/writing a local variable or common variable, use a method such as the one below.

(Note 1) Local variables can be read, but cannot be written with melWriteData().

1) Retrieve Variable <Open> (equivalent to #0)

Create a program (ex. Visual Basic) such as List 1. , and then retrieve <Open> value beforehand. For Visual Basic, call the List 1 function during loading of the form, retrieve <Open> data in the dNullData variable, and then read or write the variable using List 2 and List 3 functions.
Section No. 29 and sub-section No. 20000 become numbers retrieving an <Open> value.

List 1.

```
Global dNullData As Double              ' This is a variable for storing <Open> data

' Retrieve <Open> data information, and then save in dNullData.
' Check the <Open> data with this variable.
```

```
Function fnGetNullData (lNCDriveNumber As Long) As Long
    Dim dReadData As Double
    Dim lAddress As Long
    Dim lSectionNum As Long
    Dim lSubSectionNum As Long
    Dim lReadType As Long
    Dim dwStatus As Long

    lAddress = ADR_MACHINE(lNCDriveNumber)
    lSectionNum = 29              ' Section No. 29
    lSubSectionNum = 20000        ' Sub-section No. 2000
    lReadType = T_DOUBLE
    dwStatus = melReadData (Me.hWnd, lAddress, lSectionNum, lSubSectionNum, dReadData,
    lReadType)
    If RetvIsError (dwStatus) = True Then
        dNullData = 0
    Else
        dNullData = dReaddata
    End If
    fnGetNullData = dwStatus
End Function
```

2) When reading common variables (#100~, #500~), retrieve the variables from a program such as List 2. Compare with dNullData and determine if the variable is <Open> or not. If the variable is <Open>, substitute a space, if the variable is a zero, substitute "0."

List 2. Reading the common variables

```
Function fnReadCommonVar (lNCDriveNumber As Long, lSystemNum As Long, lCommonNm As
Long) As String
    Dim lAddress As Long
    Dim lSectionNum As Long
    Dim lSubSectionNum As Long
    Dim lReadtype As Long
    Dim dwStatus As Long
    Dim dReadData As Double

    ' Read the common variable data selected from the NC Card.
    lAddress = ADR_MACHINE (lNCDriveNumber) Or ADR_SYSTEM (lSystemNum)
    If lCommonNm < 500 Then
        lSectionNum = 32  ' #100~ : Section No. 32
    Else
        lSectionNum = 29  ' #500~ : Section No. 29
    End If
    lSubSectionNum = 20000 + lCommonNm
    lReadType = T_DOUBLE

    dwStatus = melReadData (Me.hWnd, lAddress, lSectionNum, lSubSectionNum, dReadData,
    lReadType)
    If RetvIsError (dwStatus) = True Then
        fnReadCommonVar = "Read Error."
    End If

    ' If data is <Open>, return a blank space.
    If (dReadData = dNullData) Then
        fnReadCommonVar = ""
    Else
        fnReadCommonVar = Str$ (dReadData)
    End If

End Function
```

List 3. Writing the common variables

```
Function fnWriteCommonVar (lNCDriveNumber As Long, lSystemNum As Long, lCommonNm As
Long, sWriteData As String) As Long
    Dim dWriteData As Double
    Dim lAddress As Long
    Dim lSectionNum As Long
    Dim lSubSectionNum As Long
    Dim lWriteType As Long
    Dim dwStatus As Long

    If (RTrim$ (sWriteData) = "") Then
        ' If the setting part of common variable data is NULL, then set the data to <Open>.
        dWriteData = dNullData
    Else
        ' Value convert the data in the setting part of the common variable data.
        dWriteData = CDbl (sWriteData)
    End If

    ' Write the input common variable data in the NC Card.
    lAddress = ADR_MACHINE (lNCDriveNumber) Or ADR_SYSTEM (lSystemNum)
    If lCommonNm < 500 Then
        lSectionNum = 32            ' #100~ : Section No. 32
    Else
        lSectionNum = 29            ' #500~ : Section No. 29
    End If
    lSubSectionNum = 20000 + lCommonNum
    lWriteType = T_DOUBLE
    dwStatus = melWriteData (Me.hWnd, lAddress, lSectionNum, lSubSectionNum, dWriteData,
    lWriteType)

    fnWriteCommonVar = dwStatus

End Function
```

4.2 Reading and writing machine error compensation parameters

Read/write the #4101~#5124 machine error compensation amount parameters using the following numbers.

Section No.= 3
Sub-section No. = 384 + (n–4101) n=4101~5123 (**Always designate an odd number in n**)
Read/write the data as a pair of compensation amounts.

Example

#4101 = 3
#4102 = 5
#4103 = 7
#4104 = 10
In this case, when section No. 3 and sub-section No. 384 is designated and read, &H0305 is retrieved as the return value of melReadData. When section No. 3 and sub-section No. 386 is designated and read, &H070A is retrieved as the return value of melReadData.

The above values are explained in the following example.
When rewriting #4104, first retrieve #4103 and #4104 with melReadData (&H070A is retrieved). Next, take the retrieved value and the AND (logical AND) of &HFF00, and mask the low-order 1-byte (becomes &H0700). Lastly, add the rewrite value (for example &H12), and write &H0712 with melWriteData.
The program below is an example of a function procedure to set the machine error compensation amount. Set and call the compensation data setting number (number equivalent to #4101-#5124) in lCompNum, and the compensation data to be stored in lSetData.

Writing the machine error compensation amount

```
Function fnWriteCompData (lNCDriveNumber As Long, lCompNum As Long, lSetData As Long) As
Long
    Dim lCompNum2 As Long              ' Sub-section No. for accessing the compensation data
    Dim lCompData As Long              ' Machining area of the compensation data
    Dim lAddress As Long
    Dim lSectionNum As Long
    Dim lSubSectionNum As Long
    Dim lpData As Long
    Dim lType As Long

    If (lcompNum And &H1) = &H0 Then
        lCompNum2 = lCompNum - 1       ' For even numbers, combine with the previous
                                         compensation amount and handle.
    Else
        lCompNum2 = lCompNum
End If

    lCompNum2 = 384 + (lCompNum2 - 4101)        ' Calculate the sub-section No. for
                                                  accessing the compensation data.
    lAddress = ADR_MACHINE (lNCDriveNumber)
    lSectionNum = 3                             ' Section No. setting
    lSubSectionNum = lCompNum2                  ' Sub-section No. setting
    lType = T_LONG                              ' lCompData data type setting
                                                ' Read the compensation data in
                                                  lCompData.
    dwStatus = melReadData (Me.hWnd, lAddress, lSectionNum, lSubSectionNum, lCompData,
    lType)
    If RetvIsError (dwStatus) =True Then
        fnWriteComopData = dwStatus
    End If
```

```
    If (lCompNum And &H1) = &H0 Then                    ' Set the compensation data in the low-
                                                          order 1-byte.
        lCompData = (lCompData And &HFF00) Or (lSetData And &HFF)
    Else                                                ' Set the compensation data in the high-
                                                          order 1-byte.
        lCompData = (lCompData And &HFF) Or (lSetData * 256)
    End If

                                                        ' Write lCompData in the NC.
    dwStatus = melWriteData (Me.hWnd, lAddress, lSectionNum, lSubSectionNum, lCompData,
    lType)
    fnWriteCompData = dwStatus

End Function
```

4.3 PLC device access

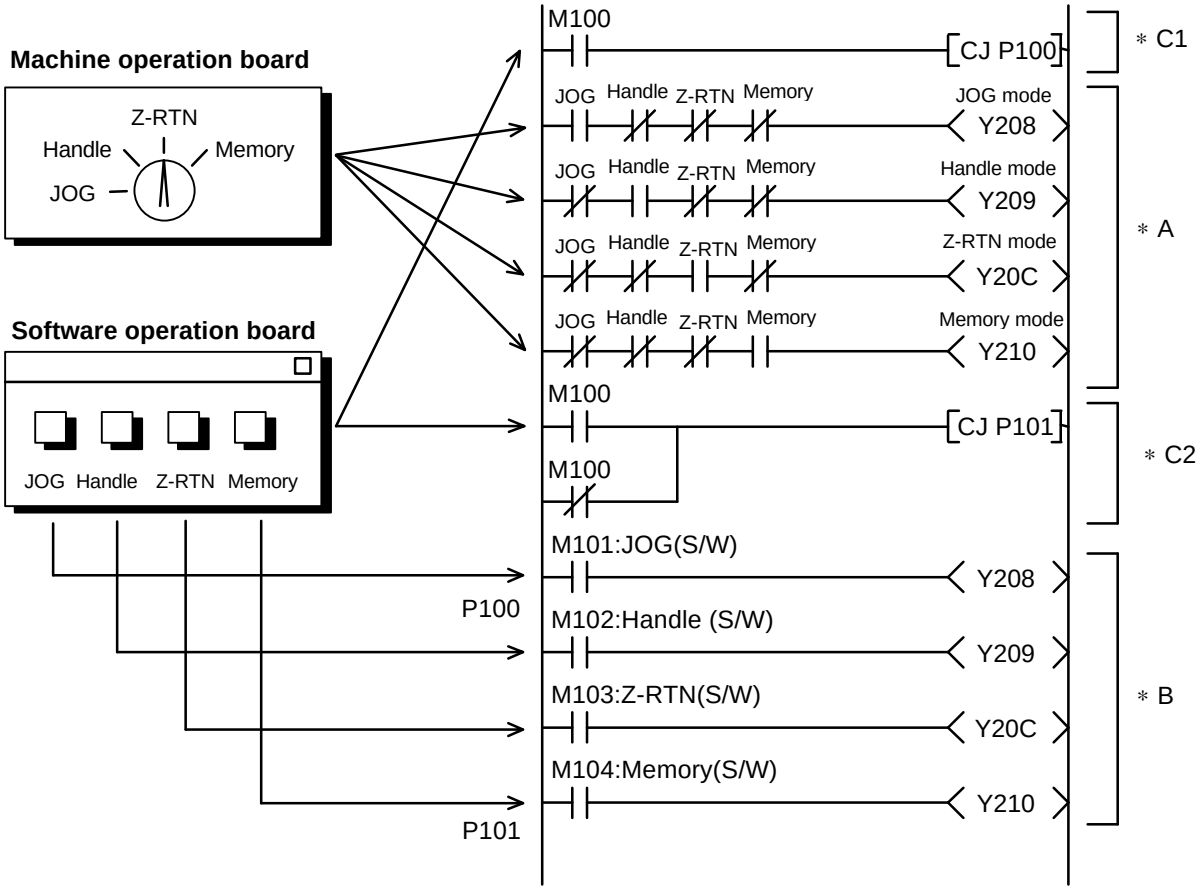
The PLC device is classified by section No. in each type of device (X, Y, D, R, etc.), or the device numbers are classified by sub-section No.
The bit device normally reads/writes using the section Nos. 10002~10025.
The following program is an example reading X8 and writing 1 in Y210.

```
Dim lAddress As Long
Dim lSectionNum As Long
Dim lSubSectionNum As Long
Dim lDevice As Long                 ' For X device storage
Dim lType As Long                   ' For data type storage of lDevice variable

lAddress = ADR_MACHINE(1)

' Read/write X8
lSectionNum = 10024                 ' Section No. setting
lSubSectionNum = &H8                ' Sub-section No. setting
lType = T_LONG                      ' Set variable lDevice data type
dwStatus = melReadData (Me.hWnd, lAddress, lSectionNum, lSubSectionNum, lDevice,
lType)
If RetvIsError (dwStatus) = True Then
    ' Describe error process
End If

' Write 1 in Y210
lSectionNum = 10025                 ' Section No. setting
lSubSectionNum = &H210              ' Sub-section No. setting
lType = T_LONG                      ' Set variable lDevice data type
lDevice = 1
dwStatus = melWriteData (Me.hWnd, lAddress, lSectionNum, lSubSectionNum, lDevice,
lType)
If RetvIsError (dwStatus) = True Then
    ' Describe error process
End If
```

The above writing method to the PLC is called one-shot writing, because the data is set for the designated device only once when written with melWriteData. When creating an operation board, etc., with the software, add a circuit to process the input signals from the software, as shown in the example on the next page.

The diagram below is an example of changing over and using the operation mode changeover switch on the machine operation board to the operation board (software operation board) created with the software.

Part ∗A of the following ladder circuit is the part that processes the input of the rotary switch that changes over the operation mode of the machine operation board. Any of the modes [Y208 (JOG mode)/Y209 (Handle mode)/Y20C (Z-RTN mode)/Y210 (Memory mode)] are selected by the appropriate input (JOG/Handle/Zero point return/Memory).
Part ∗B is the part for selecting Y208 (JOG mode)/Y209 (Handle mode)/Y20C (Z-RTN mode)/Y210 (Memory mode) using inputs from the software operation board. Parts ∗C1 and ∗C2 are circuits for changing over between the machine operation board and the software operation board. When creating the software operation board, assign the software operation board valid/invalid signal (M100) and each input signal [JOG (M101)/Handle (M102)/Zero point return (M103)/Memory (M104)] to the appropriate device (here, M100~). When selecting the operation mode, turn ON M100 and validate the software operation board before selecting one-shot writing and the operation mode for the M101~ device.



Besides the one-shot writing method of writing to the PLC, there is also another method called modal writing. When written with modal writing, the NC holds the data setting information to the device, and continues to forcibly set data for the device until modal writing is canceled. (Writing in modal writing variables is equivalent to setting (Device) (Data) (2) from the MELDASMAGIC Monitor I/F Diagnosis screen. One-shot writing is equivalent to setting (Device) (Data) (1).

As described before, when modal writing is used Y208, Y209, Y20C, and Y210 can be directly controlled without creating a ladder. In other words, modal writing can be forcibly carried out in Y208~ after setting Y208~ based on the input signals of the machine operation board. If the user already has the ladder program to carry out the machine control, this method is useful in that it allows easy control of the input signals with the software. However, the range of use is restricted by the limitations below.

(1) With modal writing, a maximum of four settings to the device is valid. (If five or more modal writings is carried out, it will result in operations equivalent to one-shot writing in order from the previous settings.)

(2) The only method available for releasing modal writing releases all modal writing together.

Modal writing can be used with simple push-button control, etc., unaccompanied by a mode (it is not necessary to hold the ON/OFF status for long periods), as shown in the examples below.

• Automatic operation start button
An automatic operation start button can be created from software compatible with the machine operation board by turning ON Y218 with modal writing at a set time (approx. 0.5 sec.) When a set time passes after releasing the button, the modal writing (cancel modal writing by writing 1 in section 11000, sub-section 0) is canceled.

• Spindle forward run, reverse run/index/orient commands

• Other machine characteristic operations such as ATC tool replacement commands, etc.

⚠ CAUTION

Do not create functions with the software for avoiding machine operation danger such as the RESET button and FEED HOLD button.
Serious problems such as delays in danger avoidance may occur due to transfer lag of the software button signal.

Besides reading/writing the bit device (B, C, E, F, G, I, J, L, M, Q, T, U, X, Y) in 1-bit units, reading/writing grouped in 8-bit (1-byte), 16-bit (2-byte), or 32-bit (4-byte) units is also possible. Note that in this case the devices that can read/write continue, and are divided as below.

[When reading/writing in 1-byte units]

Numbers that are multiples of 8 such as 0, 8, 16, 24, 32, . . . (for every 1-byte) can be used in the sub-section No. For example, when read by 10124-0, 8 devices' worth (1-byte worth) of X000~X007 information can be grouped and read, and when read by 10124-8, 8 devices' worth of X008~X00F information can be grouped and read.

[When reading/writing in 2-byte units]

Numbers that are multiples of 16 such as 0, 16, 32, 48, . . . (for every 2-bytes) can be used in the sub-section No. For example, when read by 10224-16, 16 devices' worth (2-bytes worth) of X010~X01F information can be grouped and read.

[When reading/writing in 4-byte units]

Numbers that are multiples of 32 such as 0, 32, 64, 96 . . . (for every 4-bytes) can be used in the sub-section No. For example, when read by 10324-32, 32 devices' worth (4-bytes worth) of X020~X03F information can be grouped and read.

In reading/writing with 1, 2, and 4-byte units, numbers outside byte divisions as 1, 2, and 10 cannot be used as above in the sub-section Nos.

The section Nos. for inputting/outputting 1-byte, 2-byte, and 4-byte units are shown.

10102 ~ 10125	1-byte unit (one-shot)
10202 ~ 10225	2-byte unit (one-shot)
10302 ~ 10325	4-byte unit (one-shot)

Revision History

Sub-No.	Date of revision	Revision details
*	July, 1997	First edition created.

MITSUBISHI ELECTRIC CORPORATION
HEAD OFFICE: MITSUBISHI DENKI BLD. MARUNOUCHI. TOKYO 100 TEL:03-218-3426