OMRON

Programmable Terminal

NS-B-series

Macro Reference Manual

NB15W-TW01B-V1-S
NB10W-TW01B-V1-S
NB7W-TW01B-V1-S
NB5Q-TW01B-V1-S



V132-E1-01

NOTE

- All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, mechanical, electronic, photocopying, recording, or otherwise, without the prior written permission of OMRON.
- No patent liability is assumed with respect to the use of the information contained herein. Moreover, because OMRON is constantly striving to improve its high-quality products, the information contained in this manual is subject to change without notice.
- Every precaution has been taken in the preparation of this manual. Nevertheless, OMRON assumes no responsibility for errors or omissions.
  Neither is any liability assumed for damages resulting from the use of the information contained in this publication.

Trademarks

- Sysmac and SYSMAC are trademarks or registered trademarks of OMRON Corporation in Japan and other countries for OMRON factory automation products.
- Microsoft, Windows, Microsoft Edge, and Excel are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.
- EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.
- ODVA, CIP, CompoNet, DeviceNet, and EtherNet/IP are trademarks of ODVA.
- The SD and SDHC logos are trademarks of SD-3C, LLC.

Other company names and product names in this document are the trademarks or registered trademarks of their respective companies.

Copyrights

Microsoft product screen shots used with permission from Microsoft.

Introduction

Thank you for purchasing an NS-B-series Programmable Terminal.

This manual contains information that is necessary to use the NS-B-series Programmable Terminal. Please read this manual and make sure you understand the functionality and performance of the NS-B-series Programmable Terminal before you attempt to use it in a control system.

Keep this manual in a safe place where it will be available for reference during operation.

Based on the assumption that many users will use the NS-B-series PT as a replacement for their NS-series PT, the products other than PT that may have already been discontinued are described in this manual. If you are considering purchasing these products, check that the relevant product has not been discontinued.

Intended Audience

This manual is intended for the following personnel, who must also have knowledge of electrical systems (an electrical engineer or the equivalent).

- Personnel in charge of introducing FA systems.
- Personnel in charge of designing FA systems.
- Personnel in charge of installing and maintaining FA systems.
- Personnel in charge of managing FA systems and facilities.

Applicable Products

This manual covers the following products.

- NS-B-series Programmable Terminal
  NB15W-TW01B-V1-S, NB10W-TW01B-V1-S, NB7W-TW01B-V1-S, and NB5Q-TW01B-V1-S

Relevant Manuals

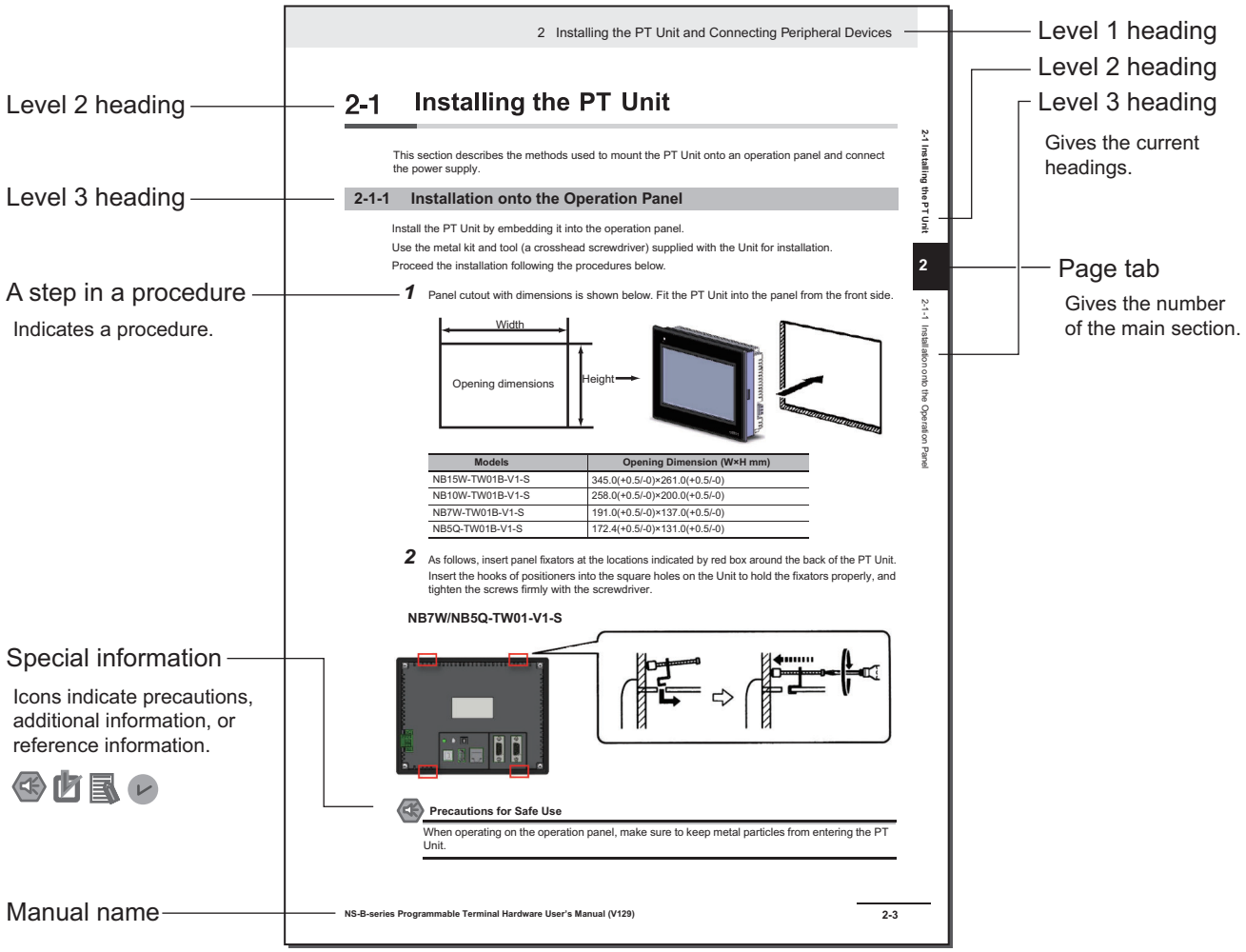
The basic information required to use an NS-B-series PT is provided in the following manuals.

- *NS-B-series Programmable Terminal Hardware User's Manual (Cat. No. V129)*
- *NS-B-series Programmable Terminal Software User's Manual (Cat. No. V130)*
- *NS-B-series Programmable Terminal Host Connection User's Manual (Cat. No. V131)*
- *NS-B-series Programmable Terminal Macro Reference Manual (Cat. No. V132)*
- *CX-Designer Version 4.□ Operation Manual (Cat. No. V133)*

Manual Structure

Page Structure

The following page structure is used in this manual.



This illustration is provided only as a sample. It may not literally appear in this manual.

Special Information

Special information in this manual is classified as follows:



Precautions for Safe Use

Precautions on what to do and what not to do to ensure safe usage of the product.



Precautions for Correct Use

Precautions on what to do and what not to do to ensure proper operation and performance.



Additional Information

Additional information to read as required.

This information is provided to increase understanding or make operation easier.



Version Information

Information on differences in specifications and functionality for different versions of the CX-Designer is given.

Precaution on Terminology

In this manual, “download” refers to transferring data from the CX-Designer to the PT Unit and “upload” refers to transferring data from the PT Unit to the CX-Designer.

Sections in this Manual

1 Outline of Macro Function

2 Functions

3 Error Message List

I Index

CONTENTS

Introduction 1
Intended Audience 1
Applicable Products 1

Relevant Manuals 2

Manual Structure 3
Page Structure 3
Special Information 4
Precaution on Terminology 4

Sections in this Manual 5

Terms and Conditions Agreement 8

Safety Precautions 9

Precautions for Safe Use 10

Precautions for Correct Use 11

Regulations and Standards 12

Related Manuals 13

Terminology 14

Revision History 15

Section 1 Outline of Macro Function

1-1 What is Macro? 1-2

1-2 Macro Execution Condition 1-3
1-2-1 Executing conditions for the project 1-3
1-2-2 Execution conditions for the screen 1-3
1-2-3 Execution conditions for the functional objects 1-4

1-3 Macro Programming 1-6
1-3-1 The Method for Writing a Macro 1-6
1-3-2 Programming Terms 1-7

Section 2 Functions

2-1 Table of Function and Argument 2-2

2-2 Details of the Function 2-5

2-3 Address Type Number 2-38

Section 3 Error Message List

3-1 Error Message List 3-2

Index

Terms and Conditions Agreement

WARRANTY

- The warranty period for the Software is one year from the date of purchase, unless otherwise specifically agreed.
- If the User discovers defect of the Software (substantial non-conformity with the manual), and return it to OMRON within the above warranty period, OMRON will replace the Software without charge by offering media or download from OMRON's website. And if the User discovers defect of media which is attributable to OMRON and return it to OMRON within the above warranty period, OMRON will replace defective media without charge. If OMRON is unable to replace defective media or correct the Software, the liability of OMRON and the User's remedy shall be limited to the refund of the license fee paid to OMRON for the Software.

LIMITATION OF LIABILITY

- THE ABOVE WARRANTY SHALL CONSTITUTE THE USER'S SOLE AND EXCLUSIVE REMEDIES AGAINST OMRON AND THERE ARE NO OTHER WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTY OF MERCHANTABILITY OR FITNESS FOR PARTICULAR PURPOSE. IN NO EVENT, OMRON WILL BE LIABLE FOR ANY LOST PROFITS OR OTHER INDIRECT, INCIDENTAL, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF USE OF THE SOFTWARE.
- OMRON SHALL HAVE NO LIABILITY FOR DEFECT OF THE SOFTWARE BASED ON MODIFICATION OR ALTERNATION TO THE SOFTWARE BY THE USER OR ANY THIRD PARTY. OMRON SHALL NOT BE RESPONSIBLE AND/OR LIABLE FOR ANY LOSS, DAMAGE, OR EXPENSES DIRECTLY OR INDIRECTLY RESULTING FROM THE INFECTION OF OMRON PRODUCTS, ANY SOFTWARE INSTALLED THEREON OR ANY COMPUTER EQUIPMENT, COMPUTER PROGRAMS, NETWORKS, DATABASES OR OTHER PROPRIETARY MATERIAL CONNECTED THERETO BY DISTRIBUTED DENIAL OF SERVICE ATTACK, COMPUTER VIRUSES, OTHER TECHNOLOGICALLY HARMFUL MATERIAL AND/OR UNAUTHORIZED ACCESS.
- OMRON SHALL HAVE NO LIABILITY FOR SOFTWARE DEVELOPED BY THE USER OR ANY THIRD PARTY BASED ON THE SOFTWARE OR ANY CONSEQUENCE THEREOF.

APPLICABLE CONDITIONS

USER SHALL NOT USE THE SOFTWARE FOR THE PURPOSE THAT IS NOT PROVIDED IN THE ATTACHED USER MANUAL.

CHANGE IN SPECIFICATION

The software specifications and accessories may be changed at any time based on improvements and other reasons.

ERRORS AND OMISSIONS

The information in this manual has been carefully checked and is believed to be accurate; however, no responsibility is assumed for clerical, typographical, or proofreading errors, or omissions.

Safety Precautions

Refer to the following manuals for safety precautions.

- *NS-B-series Programmable Terminal Hardware User's Manual (Cat. No. V129)*

Precautions for Safe Use

Refer to the following manuals for precautions for safe use.

- *NS-B-series Programmable Terminal Hardware User's Manual (Cat. No. V129)*

Precautions for Correct Use

Refer to the following manuals for precautions for correct use.

- *NS-B-series Programmable Terminal Hardware User's Manual (Cat. No. V129)*

Regulations and Standards

Refer to the following manuals for regulations and standards.

- *NS-B-series Programmable Terminal Hardware User's Manual (Cat. No. V129)*

Related Manuals

The followings are the manuals related to this manual. Use these manuals for reference.

Manual name	Cat. No.	Models	Applications	Description
NS-B-series Programmable Terminal Hardware User's Manual	V129	NB□□□-TW01B-V1-S	Learning the specifications and settings required to install an NS-B-series PT and connect peripheral devices.	Information is provided on NS-B-series Programmable Terminal specifications, part names, installation procedures, and procedures to connect an PT Unit to peripheral devices. Information is also provided on maintenance after operation and troubleshooting.
NS-B-series Programmable Terminal Software User's Manual	V130	NB□□□-TW01B-V1-S	Learning about NS-B-series PT pages and object functions.	Describes the pages and object functions of the NS-B-series Programmable Terminals.
NS-B-series Programmable Terminal Host Connection User's Manual	V131	NB□□□-TW01B-V1-S	Learning the specifications required to connect devices to an NS-B-series PT.	Information is provided on connection procedures and setting procedures to connect an NS-B-series PT to a Controller or other device.
NS-B-series Programmable Terminal Macro Reference Manual	V132	NB□□□-TW01B-V1-S	When you wish to know the language specifications of the macros used with the NS-B-series PT.	Describes the language specifications and functions of the macros used with the NS-B-series PT.
CX-Designer Version 4.□ Operation Manual	V133	CXONE-AL□□C-V4 CXONE-AL□□D-V4	When you wish to know how to create screens in CX-Designer and how to download to the PT Unit.	Describes how to create screens in CX-Designer and the procedure to download to the PT Unit.

Refer to the following manual for other related manuals.

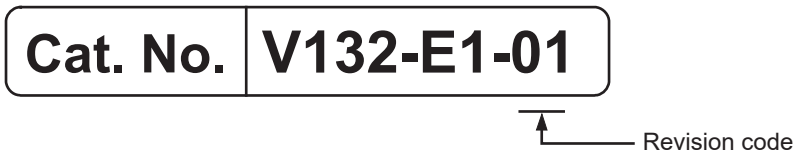
- *NS-B-series Programmable Terminal Hardware User's Manual (Cat. No. V129)*

Terminology

Term	Description
PT Unit	In this manual, indicates an NS-B-series Programmable Terminal.
PLC	Indicates Programmable Controllers in the OMRON SYSMAC CS/CJ, C, or CVM1/CV Series of Programmable Controllers.
CS Series	Indicates Programmable Controllers in the OMRON SYSMAC CS Series of Programmable Controllers: CS1G, CS1H, CS1G-H, or CS1H-H
C Series	Indicates Programmable Controllers in the OMRON SYSMAC C Series of Programmable Controllers: C200HS, C200HX/HG/HE-E/-ZE, CQM1, CQM1H, CPM1A, CPM2A, CPM2C
CVM1/CV Series	Indicates Programmable Controllers in the OMRON SYSMAC CV/CVM1 Series of Programmable Controllers: CV500, CV1000, CV2000, CVM1
CJ Series	Indicates Programmable Controllers in the OMRON SYSMAC CJ Series of Programmable Controllers: CJ1G, CJ1G-H, CJ1H-H, CJ1M, CP1H, CP1L, CP1E, CP2E, CJ2H, CJ2M References to “CJ1H-H” in this manual indicate the CJ1H-CPU□□H-R and CJ1H-CPU□□H. References to “CJ1H-H-R” indicate the CJ1H-CPU□□H-R. “CJ2” is used when referring specifically to only the CJ2 CPU Units.
NJ Series	Indicates a Controller in the OMRON NJ Series of Controllers: (Only models that can be connected to NS-B-series PTs are given.): NJ101, NJ301, NJ501
NX Series	Indicates Controllers in the OMRON NX Series of Controllers: (Only models that can be connected to NS-B-series PTs are given.): NX701, NX102, or NX1P2
Serial Communications Unit	Indicates a Serial Communications Unit for an OMRON CS/CJ-series PLC.
Serial Communications Board	Indicates a Serial Communications Board for an OMRON CS-series or CQM1H PLC.
Communications Board	Indicates a Communications Board for an OMRON C200HX/HG/HE-E/-ZE PLC.
CPU Unit	Indicates a CPU Unit in the OMRON SYSMAC CS/CJ, C, or CVM1/CV Series of Programmable Controllers.
CX-Designer	Indicates the OMRON CX-Designer.
Host	Indicates the PLC, FA computer, or personal computer functioning as the control device and interfaced with the NS-B-series PT.
Removable media	Indicates a USB memory.
Project	Indicates a collection of files containing screen data, settings, and other information for the NS-B-series that are created in CX-Designer.

Revision History

A manual revision code appears as a suffix to the catalog number on the front and back covers of the manual.



Revision code	Date	Revised content
01	October 2026	Original production

1

Outline of Macro Function

This section describes execution conditions and programming procedure for using macro.

1-1 What is Macro? 1-2

1-2 Macro Execution Condition 1-3

1-2-1 Executing conditions for the project 1-3

1-2-2 Execution conditions for the screen 1-3

1-2-3 Execution conditions for the functional objects 1-4

1-3 Macro Programming 1-6

1-3-1 The Method for Writing a Macro 1-6

1-3-2 Programming Terms 1-7

1-1 What is Macro?

Macro is the function which can be executed by users original program. User can add functions, such as arithmetic operations and distinction of conditions, which are not supported by standard functions in CX-Designer. This function allows the PT Unit to process screen display or data, which is performed by PLC before.It is also possible to reduce a load of PLC.

In this manual, the timing for executing macro is called “Macro Execution Condition”.

Macro can be made by roughly divided three execution conditions as shown below.

- Execution condition for the project
- Execution condition for the screen
- Execution condition for the functional objects

There is no restriction on the number of macros for 1 project/1 screen.

Up to 3000 characters can be used for one macro. Line feed is counted as two characters. There is also no restriction on the number of lines.

Example:

‘Number of inputting characters	....11 characters + line feed (2 characters), including comment
$W0=10;	....7 characters + line feed (2 characters)
STRCPY($W10”,ABCDE”);	....21 characters

In this case, 43 characters are used.

1-2 Macro Execution Condition

Macros can be created for each project, screen, and functional objects. Also they can be created for the following execution conditions.

1-2-1 Executing conditions for the project

Macro execution conditions that can be made for the project are indicated below.

Select [PT]-[Project Properties]-[Macro] tab in CX-Designer, then set execution condition and record macro.

For details on registering macros, refer to the online CX-Designer Help ‘System Settings and Project Properties’.

Execution condition	Explanation
When Loading a Project	Execute just before loading the first screen after starting up PT Unit
ON timing Alarm/Event occurred	Execute when alarm is occurred
ON timing Alarm/Event is canceled	Execute when alarm is cancelled
When a bit changed	Set macro to execute when the address of specified bit type is changed. Up to 10 macros can be set.
When a value changed	Set macro to execute when the address of specified word type is changed. Up to 10 macros can be set.

1-2-2 Execution conditions for the screen

Macro execution conditions that can be made for each screen are indicated below.

Select [PT]-[Screen/Sheet Properties]-[Macro] tab screen in CX-Designer, then set execution condition and record macro.

For details on registering macros, refer to the online CX-Designer Help ‘Creating Screens’.

Execution condition	Explanation
When Loading a Screen	Execute immediately after reading screen data to display the next
When Unloading a Screen	Execute immediately after closing the current screen



Additional Information

Macros are executed by the timing as shown below.

	When loading a screen (Execute at the destination screen)	When unloading a screen (Execute at the destination screen)
User screen→User screen	Executed	Executed
User screen→Transfer screen	Not Executed	Executed
User screen→System menu	Not Executed	Executed
System menu→User screen	Executed	Not Executed
User screen→Screen Saver	Not Executed	Not Executed
Screen Saver→User screen	Not Executed	Not Executed

1-2-3 Execution conditions for the functional objects

Macro execution conditions that can be made for each functional object are described in the following table.

Open property dialog for each functional object and select [Macro] tab page, then set the execution condition and create macro.

For details on registering macros, refer to the online CX-Designer Help ‘Creating Functional Objects’.

Execution condition	Explanation
Touch on Timing	Execute when functional object is pressed.
Touch off Timing	Execute when functional object is released.
Before Inputting numeral or character string	Execute just before display tenkey pad or virtual keyboard for inputting values or strings.
Before Writing numeral or character string	Execute just before notice numeral and character string to the host.
When changing numeral and character string and comparing numeral	Execute when changing the value of address.
When Processing Display Area	Execute when display area for alarm display is pressed.
When Selecting an Alarm/Event	Execute just after select each alarm/event displayed on Alarm/Event Summary.
When selecting a list	Execute just after select a list displayed on the List Selection.

Precautions for Correct Use

If the password is set for functional objects, the following macros are executed after inputting the password. If the password has been cancelled for inputting, macros will not be executed.

- Touch on timing
- Touch off timing
- Before inputting Numeral/Character string
- When pressing Display Area
- When selecting an Alarm/Event
- When selecting a list

The following conditions can be selected for functional objects.

Functional Object	Touch on Timing	Touch Off Timing	When changing Numeral/ String and comparing Numeral	Before inputting Numeral/ String	Before writing Numeral/ String	List Selection
ON/OFF Button	OK	OK	---	---	---	---
Word Button	OK	OK	---	---	---	---
Command Button	OK	OK	---	---	---	---
Bit Lamp	---	---	OK	---	---	---
Word Lamp	---	---	OK	---	---	---
Numeral Display & Input	---	---	OK	OK	OK	---
String Display & Input	---	---	OK	OK	OK	---
Thumbwheel Switch	---	---	OK	---	OK	---
Text	---	---	---	---	---	---
List Selection	---	---	---	---	---	OK
Level Meter	---	---	---	---	---	---

Functional Object	Touch on Timing	Touch Off Timing	When changing Numeral/ String and comparing Numeral	Before inputting Numeral/ String	Before writing Numeral/ String	List Selection
Broken-line Graph	---	---	---	---	---	---
Bitmap	---	---	---	---	---	---
Analogue Meter	---	---	---	---	---	---
Video Display	---	---	---	---	---	---
Date	---	---	---	---	---	---
Time	---	---	---	---	---	---
Data Log Graph	---	---	---	---	---	---
Data Block Table	---	---	---	OK	OK	---
Temporary Input	---	---	---	---	---	---
Consecutive line drawing	---	---	---	---	---	---
Multifunction Object	OK	OK	OK	---	---	---
Contents Display	---	---	---	---	---	---

[Alarm/Event object]

Functional Object	When Pressing a Display Area	When selecting an Alarm/Event
Alarm/Event Display	OK	---
Alarm/Event Summary	---	OK



Additional Information

When a command button is set in the following functions, macros that are set at the touch on will be executed at the touch off. Macros that are set at the off will not be executed.

- Switch screen
- Control pop-up screen
- Display system menu
- Data block control

When functions other than above are selected, macros will be executed at the specified timing.

1-3 Macro Programming

This section describes macro creation procedures and programming terms.

1-3-1 The Method for Writing a Macro

Delimiter of the Program

Put a semicolon (;) at the end of each program as a delimiter. However, it is not necessary for IF(), ELSEIF(), ELSE(), ENDIF.

Example:

```
$W0=2;
IF($W0>=10)
    $W5=$W0-$W2;
ELSE
    $W5=$W0+$W2;
ENDIF
```

Comment

Put single quotation mark at the beginning of the sentence when you add the comment for each program. From single quotation mark (‘) to the end of character sting will be regarded as a comment.

Example:

```
$W0=100; ‘Comment
‘Comment
IF($W1==200)
```

Writing Programming Terms

Both uppercase and lowercase can be used for programming macros because they are not classified.

Furthermore, even if a mixture of uppercase and lowercase are used to make the macros easy to understand, they will be processed normally.

However, if a double-byte character is used for a tag, the tag is case-sensitive and width-sensitive.

Example:

- MovePopwDown() and MOVEPOPWDOWN() are regarded as same function.
- [Host1:DM0] and [host1:dm0] are regarded as same host address.
- [Host1:TAG1](single-byte), [host1:ＴＡＧ１](double-byte) and [host1:ｔａｇ１](double-byte) are regarded as different host addresses.

1-3-2 Programming Terms

This section describes terms used in this function.

Variable

The following variables can be used in macro program.

Item	Explanation
Host address	Enclose address and tag in [ ]. Host address is used for functions (READCMEM and WRITECMEM) for communicating with a host. Example: READCMEM($W100,[HOST1:DM00000],100); 'Read HOST1:DM00000 to DM00099 to $W100 to $W199
PT Memory	Bit Internal memory: $B $B0 to $B32767 (1bit per 1point) Internal Holding memory: $HB $HB0 to $HB8191 (1bit per 1point) System Memory: $SB $SB0 to $SB 63 (1bit per 1point) Word Internal memory: $W $W to $W32767 (16bit per 1point) Internal Holding memory: $HW $HW0 to $HW 8191 (16bit per 1point) System Memory: $SW $SW0 to $SW39 (16bit per 1point) Example: $W100=$W0+1; 'Set the value $W0 and 1 added to $W100
Index	Index is used for processing bit and word in the PT memory. Add index to the end of the address and it will processed as [specified address + index value]. There are 10 index points (I0 to I9). Set I0 to I9 for the value of $SW27 to $SW36. Example: $SW27=H20; $W0I0=123; '$W0I0 is regarded as $W20 added $W0 and 20 '$W20=123

Qualifier of Variable

Qualifiers set for variables must be used as shown below.

Qualifiers are used when performing 32-bit data processing and numeral processing for bit.

Item	Explanation
Long Access (32bit) of Word (16bit)	Put "L" at the end of variable. Uses 2 words. $W0L=1000000; 'Accesses regarding $W0, $W1 as 32-bit $W100L=1000*1000; 'Accesses regarding $W100, $W101 as 32-bit
Numeral Access of Bit	Put ":n" at the end of variable. Specify the value of bit address (up to 32 by 4-bit unit) for "n". Exceptions: If n=16, input "W". If n=32, input "L". $B0:4=3; 'Sets 3(0011) for 4-bit from $B0 to $B3 $B0W=12345; 'Sets 12345(0011000000111001) for 16-bit from $B0 to '$B15

Constant

Usable constants for macro program and procedure are described in the following table.

Item	Explanation
Decimal constant	32768 to 32767 can be input when using word (16-bit) 2147483648 to 2147483647 can be input when using 2 words (32-bit)
Hexadecimal constant	H0 to HFFFF can be input when using word (16-bit) H0 to HFFFFFFFF can be input when using 2 words (32-bit)
Character String	Enclose in “ ” E.g. “ABCDE”

Branches

The following keywords can be used for specifying conditions.

Item	Explanation

IF
ELSEIF
ELSE
ENDIF
	Enclose conditional expressions in parentheses () after the IF and ELSEIF. Always use ENDIF at the end. Up to 8 nests can be input.
	There is no restrictions for inputting the number of lines under IF sentence. However, the total number of characters used in whole macro must be no more than 3000 characters.
	E.g.
	IF($W100==1) ‘If $W100 is 1
	$W99=1;
	ELSEIF($W100==2) ‘if $W100 is 2
	$W99=2;
	ELSE ‘if $W100 is other than 1 or 2
	$W99=3;
	ENDIF

Conditional Expressions

Use the following conditional expressions for specifying conditions in the IF sentences. It works for all types of data (word, long access of word, bit, and numeral access of bit).

Item	Explanation
A==B	If A is equal to B, TRUE.
A>B	If A is greater than B, TRUE.
A>=B	If A is greater than or equal to B, TRUE.
A<B	If A is less than B, TRUE.
A<=B	If A is less than or equal to B, TRUE.
A<>B A!=B	If A is not equal to B, TRUE.
A&&B A AND B	If both A and B are true, TRUE. (AND)
A||B A OR B	If either of A or B is true, TRUE. (OR)

Additional Information

- The result (A&&B, A>B) can be substituted for the variable.

  E.g. $B100=$W0>100;

  If the value of $W is “100” or less, “0” will be substituted for $B100. If the value of $W is more than “100”, “1” will be substituted for $B100.
- Index can be used.

  E.g.

```
$SW27=H100;      ‘$W100 is placed in $W0i0, as 100 is added to $W0.
  $W0i0=120;     ‘$W100=120
IF($W0i0>100)    ‘If the value for $W100 is 101 or more,
  $B0i0=1;
ELSE             ‘If the value for $W100 is 100 or less,
  $B0i0=0;
ENDIF
```

- Host Side Address can not be used.

Basic Operational Statements

The following operational statements can be used in the program.

Item	Operator	Example	Meaning
substitution	=	A=B	Substitute B for A
addition	+	C=A+B	Set A+B to C
subtraction	-	C=A-B	Set A-B to C
multiplication	*	C=A*B	Set A×B to C
division	/	C=A/B	Set A/B to C
residue	%	C=A%B	Set A%B to C
OR	|	C=A|B	Logically ORs for A and B
AND	&	C=A&B	Logically ANDs for A and B
NOT	!	C=!A	Set C for denial of A
Exclusive	^	C=A^B	Result C of logical XORs for A and B
Complement of 1	~	B=~A	Set the 1-complement of A to B
Bit Shift (left)	<<	C=A<<B	Set the value that A is Itic shifted B-bit to the left to C.
Bit Shift (right)	>>	C=A>>B	Set the value that A is arithmetic shifted B-bit to the right to C.

Additional Information

If executing logical operation, process must be performed between the same data types (between word, bit, or long access).

E.g. $SW0L=$SW10L&$W20L; ‘Use all long access of word’

Multiple operations can be combined.

E.g. A=(B+C)*(D+E/2)

The priority of orders is as follows.

Item	Symbol
High	( )
	~, !
	*, /, %
	+, -
	<<, >>
	&
	^
	|
Low	=

Functions

The following functions are provided for macro.

Action		Function
Conversion between BCD and BIN	Value (BIN code)→BCD code	BCD
	BCD code→value (BIN code)	BIN
Manipulating character strings	Copy character string	STRCPY/STRCPYW
	Convert from ASCII code to Unicode	STRM2W
	Convert from Unicode to ASCII code	STRW2M
Alarm/Event summary	Clear the number of Alarm/Event occurrence	RSTALARMCNT
HMI exclusive statements	Output written value and changed value	GETNUMVAL
	Switch screen	SHOWPAGE/ SHOWPAGEBCD
	Movement of object display area	MOVEPARTS
	Display message dialog box	MSGBOX
	Get displayed rectangle of the object	GETPARTS
	Move pop-up window	MOVEPOPW
	Move pop-up window up	MOVEPOPWUP
	Move pop-up window down	MOVEPOPWDOWN
	Move pop-up window left	MOVEPOPWLEFT
	Move pop-up window right	MOVEPOPWRIGHT
	Close pop-up window	CLOSEPOPW
Communications	Read data from specified address	READCMEM
	Write data to specified address	WRITECMEM
	Read bit data from specified address	READHOSTB
	Read word data from specified address	READHOSTW
	Write bit data to specified address	WRITEHOSTB
	Write word data to specified address	WRITEHOSTW
Process termination	Terminate macro program	RETURN
Set Date/Time	Change settings of internal clock of the PT Unit	SETTIME
Reading/writing data	Read contents (values in binary) of the specified file in a removable media to PT memory	READCF
	Save the contents of PT memory in a removable media	WRITECF

Action		Function
Write to multiple addresses	Write (0/1) to multiple bit addresses in the PT memory	BITSET
	Write a value to multiple word addresses in the PT memory	MEMSET
Data manipulation/ conversion	Swap high order and low order of the specified address	SWAP
	Swap high order (2byte) and low order (2byte) of the specified long word data	SWAPL
	Copy contents of $W in the PT memory	MEMCOPY
	Converts a local time in nanoseconds to a calendar time (year, month, day, hour, minutes, and seconds)	LOCALTIME
Input Focus Control	Set the input focus for the specified object	SETFOCUS
	Release the input focus set for the object	RELEASEFOCUS
Repeat Program	Repeat Program	FOR, NEXT
	Aborting from Program Repetition	BREAK
	Return to the top of FOR loop	CONTINUE
Numeric operations	Arccosine function	ACOS
	Arcsine function	ASIN
	Arctangent function	ATAN
	Arctangent function of two variables	ATAN2
	Ceil (Round UPWARDS) function	CEIL
	Cosine function	COS
	Converts a degree to its radian number	DEG2RAD
	Base E exponentiation function	EXP
	Floor (Round DOWNWARDS) function	FLOOR
	Natural logarithm function	LOG
	Common logarithm function	LOG10
	Converts a radian number to its degree	RAD2DEG
	Square root function	SQRT
	Sine function	SIN
	Tangent function	TAN
	Power function	POW
Floating-point arithmetic	Perform addition of floating-point values	FADD
	Perform comparison of floating-point values	FCOMP
	Perform division of floating-point values	FDIV
	Perform multiplication of floating-point values	FMUL
	Perform subtraction of floating-point values	FSUB
	Set a value converted to floating-point	FSET

The timing when a function can be executed as shown below.

	Project Macro			Screen Macro		Functional object Macro							
	When Loading a Project	Alarm/Event On Timing	Alarm/Event Off Timing	When Loading a screen	When Unloading a screen	Touch On Timing	Touch Off Timing	When changing value	Before Inputting Numeral/String	When writing Numeral/String	List Selection	When pressing a Display Area	When Selecting an Alarm/Event
ACOS	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
ASIN	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
ATAN	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
ATAN2	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
BCD	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
BIN	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
BITSET	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
BREAK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
CEIL	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
CLOSEPOPW	---	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
CONTINUE	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
COS	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
DEG2RAD	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
EXP	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
FADD	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
FCOMP	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
FDIV	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
FLOOR	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
FMUL	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
FOR, NEXT	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
FSET	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
FSUB	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
GETNUMVAL	---	---	---	---	---	---	---	△	△	△	---	---	---
GETPARTS	---	---	---	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
LOCALTIME	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
LOG	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
LOG10	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
MEMCOPY	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
MEMSET	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
MOVEPARTS	---	---	---	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
MOVEPOPW	---	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
MOVEPOPWDOWN	---	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
MOVEPOPWLEFT	---	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
MOVEPOPWRIGHT	---	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
MOVEPOPWUP	---	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
MSGBOX	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
POW	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
RAD2DEG	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
READCF	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
READCMEM	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

	Project Macro			Screen Macro		Functional object Macro							
	When Loading a Project	Alarm/Event On Timing	Alarm/Event Off Timing	When Loading a screen	When Unloading a screen	Touch On Timing	Touch Off Timing	When changing value	Before Inputting Numeral/String	When writing Numeral/String	List Selection	When pressing a Display Area	When Selecting an Alarm/Event
READHOSTB	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
READHOSTW	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
RELEASEFOCUS	---	OK	OK	---	---	OK	OK	OK	---	---	OK	OK	OK
RETURN	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
RSTALARMCNT	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
SETFOCUS	---	OK	OK	---	---	OK	OK	OK	---	---	OK	OK	OK
SETTIME	---	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
SHOWPAGE	OK	OK	OK	---	---	OK	OK	OK	OK	OK	OK	OK	OK
SHOWPAGEBCD	OK	OK	OK	---	---	OK	OK	OK	OK	OK	OK	OK	OK
SIN	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
SQRT	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
STRCPY(W)	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
STRM2W	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
SWAP	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
SWAPL	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
TAN	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
WRITECF	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
WRITECMEM	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
WRITEHOSTB	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
WRITEHOSTW	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

△: Numeral Display & Input Only

2

Functions

This section describes how to use the standard functions.

2-1 Table of Function and Argument 2-2
2-2 Details of the Function 2-5
2-3 Address Type Number 2-38

2-1 Table of Function and Argument

The variety of variables and values, which can be specified as an argument for macro function is described below. Alphabets such as S,D,n,x,y in the row indicated parameter used in *2-2 Details of the Function* on page 2-5 - “Format”.

Function	Argument	PT Memory				Constant	String	Host Side Address	Specify Index
		$B $HB $SB	Numerical access of bit	$W $HW $SW	Long access of word				
ACOS	S			Yes		Yes			
ASIN	S			Yes		Yes			
ATAN	S			Yes		Yes			
ATAN2	S			Yes		Yes			
BCD	S		Yes	Yes	Yes	Yes			Yes
BIN	S		Yes	Yes	Yes	Yes			Yes
BITSET	D	Yes							Yes
	c	Yes				Yes			Yes
	n			Yes		Yes			Yes
CEIL	S			Yes		Yes			
CLOSEPOPW	n		Yes	Yes	Yes	Yes			Yes
COS	S			Yes		Yes			
DEG2RAD	S			Yes		Yes			
EXP	S			Yes		Yes			
FADD	S1, S2			Yes		Yes			
FCOMP	S1, S2			Yes		Yes			
FDIV	S1, S2			Yes		Yes			
FLOOR	S			Yes		Yes			
FMUL	S1, S2			Yes		Yes			
FSET	S			Yes		Yes			
FSUB	S1, S2			Yes		Yes			
GETNUMVAL	None								
GETPARTS	N		Yes	Yes	Yes	Yes			Yes
	Left, Top, Right, Bottom			Yes	Yes				Yes
LOCALTIME	S, D			Yes					
LOG	S			Yes		Yes			
LOG10	S			Yes		Yes			
MEMCOPY	S			Yes	Yes				Yes
	D			Yes	Yes				Yes
	n			Yes					Yes
MEMSET	D			Yes					Yes
	c			Yes		Yes			Yes
	n			Yes		Yes			Yes
MOVEPARTS	n		Yes	Yes	Yes	Yes			Yes
	x		Yes	Yes	Yes	Yes			Yes
	y		Yes	Yes	Yes	Yes			Yes
MOVEPOPW	n		Yes	Yes	Yes	Yes			Yes
	x		Yes	Yes	Yes	Yes			Yes
	y		Yes	Yes	Yes	Yes			Yes

Function	Argument	PT Memory				Constant	String	Host Side Address	Specify Index
		$B $HB $SB	Numerical access of bit	$W $HW $SW	Long access of word				
MOVEPOPWDOWN	n		Yes	Yes	Yes	Yes			Yes
	y		Yes	Yes	Yes	Yes			Yes
MOVEPOPWLEFT	n		Yes	Yes	Yes	Yes			Yes
	x		Yes	Yes	Yes	Yes			Yes
MOVEPOPWRIGHT	n		Yes	Yes	Yes	Yes			Yes
	x		Yes	Yes	Yes	Yes			Yes
MOVEPOPWUP	n		Yes	Yes	Yes	Yes			Yes
	y		Yes	Yes	Yes	Yes			Yes
MSGBOX	S1			Yes	Yes		Yes		Yes
	S2			Yes	Yes		Yes		Yes
	S3		Yes	Yes	Yes	Yes			Yes
POW	S1, S2			Yes		Yes			
RAD2DEG	S			Yes		Yes			
READCF	Mem			Yes					Yes
	Size			Yes					
	File			Yes			Yes		
	Dev			Yes		Yes			
READCMEM	D	Yes		Yes	Yes				Yes
	S	Yes		Yes	Yes			Yes	Yes
	n					Yes			
READHOSTB	D	Yes							Yes
	h			Yes		Yes		Yes	
	ch			Yes		Yes			Yes
	addr			Yes		Yes			Yes
	r			Yes		Yes			Yes
	n			Yes		Yes			Yes
READHOSTW	D			Yes					Yes
	h			Yes		Yes		Yes	
	ch			Yes		Yes			Yes
	addr			Yes		Yes			Yes
	n			Yes		Yes			Yes
RELEASEFOCUS	None								
RETURN	S		Yes	Yes	Yes	Yes			Yes
RSTALARMCNT	S		Yes	Yes	Yes	Yes			Yes
SETFOCUS	N					Yes			Yes
SETTIME	S			Yes					
SHOWPAGE	S		Yes	Yes	Yes	Yes			Yes
SHOWPAGEBCD	S		Yes	Yes	Yes	Yes			Yes
SIN	S			Yes		Yes			
SQRT	S			Yes		Yes			
STRCPY(W)	D			Yes	Yes				Yes
	S			Yes	Yes		Yes		Yes
STRM2W	D			Yes	Yes				Yes
	S			Yes	Yes		Yes		Yes
STRW2M	D			Yes	Yes				Yes
	S			Yes	Yes		Yes		Yes
SWAP	S			Yes	Yes				Yes
	n			Yes	Yes	Yes			
SWAPL	S			Yes	Yes				
	n			Yes	Yes	Yes			

Function	Argument	PT Memory				Constant	String	Host Side Address	Specify Index
		$B $HB $SB	Numerical access of bit	$W $HW $SW	Long access of word				
TAN	S			Yes		Yes			
WRITECF	Mem			Yes					Yes
	Size			Yes					
	File			Yes			Yes		
	Dev			Yes		Yes			
WRITECMEM	D	Yes		Yes	Yes			Yes	Yes
	S	Yes		Yes	Yes				Yes
	n		Yes	Yes	Yes	Yes			
WRITEHOSTB	h			Yes		Yes		Yes	
	ch			Yes		Yes			Yes
	addr			Yes		Yes			Yes
	r			Yes		Yes			Yes
	S	Yes							Yes
	n			Yes		Yes			Yes
WRITEHOSTW	h			Yes		Yes		Yes	
	ch			Yes		Yes			Yes
	addr			Yes		Yes			Yes
	S			Yes					Yes
	n			Yes		Yes			Yes

2-2 Details of the Function

This section describes details of standard functions used in Macro programming.

ACOS

Format	ACOS(S)
Function	Returns the arccosine of variable S. S should be input as a floating-point value.
Return Value	Returns the arccosine (floating-point value). (unit: radian)
Example	$W0=ACOS($W10); 'Set the arccosine of $W10(floating-point value(2 words)) to $W0(floating-point value(2 words)). $W100=ACOS(-1.0); 'Set the arccosine of -1.0 to $W100(floating-point value(2 words)).

ASIN

Format	ASIN(S)
Function	Returns the arcsine of variable S. S should be input as a floating-point value.
Return Value	Returns the arcsine (floating-point value). (unit: radian)
Example	$W0=ASIN($W10); 'Set the arcsine of $W10(floating-point value(2 words)) to $W0(floating-point value(2 words)). $W100=ASIN(0.5); 'Set the arcsine of 0.5 to $W100(floating-point value(2 words)).

ATAN

Format	ATAN(S)
Function	Returns the arctangent of variable S. S should be input as a floating-point value.
Return Value	Returns the arctangent (floating-point value). (unit: radian)
Example	$W0=ATAN($W10); 'Set the arctangent of $W10(floating-point value(2 words)) to $W0(floating-point value(2 words)). $W100=ATAN(1.233669); 'Set the arctangent of 1.233669 to $W100(floating-point value(2 words)).

ATAN2

Format	ATAN2(S1,S2)
Function	Returns the arctangent of variable S1 and variable S2 coordinates. S1 and S2 should be input as floating-point values. S1: Y coordinates S2: X coordinates
Return Value	Returns the arctangent of two variables coordinates (floating-point value). (unit: radian)
Example	$W0=ATAN2($W10,$W20); 'Set the arctangent of $W10(floating-point value(2 words)) and $W20(floating-point value(2 words)) coordinates to $W0(floating-point value(2 words)). $W100=ATAN2(1.233669,9.258369); 'Set the arctangent of 1.233669 and 9.258369 coordinates to $W100(floating-point value(2 words)).

BCD

Format	BCD(S)
Function	Convert value "s" to BCD code. Converting range is 0 to 99999999. If you specify the character string outside of the range, overflow occurs. "0" is set at the end of the character string.
Return Value	BCD code
Example	$W0=1234; 'Set value 1234 to $W0 $W10=BCD($W0); 'Set BCD code (H1234) to $W10 $W20L=12345678; 'Set value 12345678 to $W20 to $W21 $W22L=BCD($W20L); 'Set BCD code (H12345678) to $W22 to W23

BIN

Format	BIN(S)
Function	Convert BCD code S to numeral value. Converting range is H0 to H99999999.
Return Value	Numeral value
Example	$W0=H1234; 'Set BCD code (H1234) to $W0 $W10=BIN($W0); 'Set 1234 to $W10 $W20L=H334455; 'Set BCD code (H334455) to $W20 to $W21 $W22L=BIN($W20L); 'Set 334455 to $W22 to $W23

BITSET

Format	BITSET(D,c,n)
Function	Writes c (0/1) for n-bit data from bit address D in the PT memory ($B/$HB). D: Starting address c: Set value (0/1) n: Number of elements to write 1 to 32768 ($B) 1 to 8192 ($HB)
Return Value	None
Example	• Writing 1 to 10 bits from $B100 ($B100 to $B109) BITSET($B100,1,10); • Writing 1 to 10 bits from $B100 ($B100 to $B109) $HB100=1; $W200=10; BITSET($B100,$HB100,$W200);

BREAK

Format	BREAK
Function	Interrupt a loop program between “FOR and NEXT”.
Return Value	None

Example	‘If $W100I0>30 is true, exit FOR loop.
	
	$SW27=0;
	FOR(10)
	$W100I0=$W50I0+10;
	IF($W100I0>30)
	BREAK;
	ENDIF
	$SW27=$SW27+1;
	NEXT;

CEIL

Format	CEIL(S)
Function	Returns the value of variable S rounded UPWARDS to the nearest. S should be input as a floating-point value.
Return Value	Returns the result (floating-point value).
Example	$W0=CEIL($W10); ‘Set the result of CEIL $W10(floating-point value(2 words)) to $W0(floating-point value(2 words)). $W100=CEIL(1.233669); ‘Set the result of CEIL 1.233669 to $W100(floating-point value(2 words)).

CLOSEPOPW

Format	CLOSEPOPW(n)
Function	Close pop-up window screen page number “n” Setting range for “n” is 0 to 3999. If you set pop-up screen page number that does not exist, the process will not be performed.
Return Value	None
Example	CLOSEPOPW(15); ‘Close pop-up screen page 15

CONTINUE

Format	CONTINUE
Function	During a program between “FOR to NEXT”, it will return to the top of the FOR loop and resume the FOR process.
Return Value	None

Example	‘If $W50I0>30 is true, a loop will return to the top, and resume the next repetitious program.
	
	$SW27=0;
	FOR(10)
	IF($W50I0>30)
	$SW27=$SW27+1;
	CONTINUE;
	ENDIF
	$W100I0=$W50I0+10;
	$SW27=$SW27+1;
	NEXT;

COS

Format	COS(S)
Function	Returns the cosine of variable S. S should be input as a floating-point value.
Return Value	Returns the cosine (floating-point value). (unit: radian)
Example	$W0=COS($W10); ‘Set the cosine of $W10(floating-point value(2 words)) to $W0(floating-point value(2 words)). $W100=COS(0.0); ‘Set the cosine of 0.0 to $W100(floating-point value(2 words)).

DEG2RAD

Format	DEG2RAD(S)
Function	Convert the variable S from degree to its radian and return the value. S should be input as a floating-point value.
Return Value	Returns the result of conversion (floating-point value).
Example	$W0=DEG2RAD($W10); 'Set the result of DEG2RAD $W10(floating-point value(2 words)) to $W0(floating-point value(2 words)). $W100=DEG2RAD(1.233669); 'Set the result of DEG2RAD 1.233669 to $W100(floating-point value(2 words)).

EXP

Format	EXP(S)
Function	Returns E (Euler's constant) raised to the S power. S should be input as a floating-point value.
Return Value	Returns the result (floating-point value).
Example	$W0=EXP($W10); 'Set the result of EXP $W10(floating-point value(2 words)) to $W0(floating-point value(2 words)). $W100=EXP(-0.369147); 'Set the result of EXP -0.369147 to $W100(floating-point value(2 words)).

FADD

Format	FADD(S1,S2)
Function	Returns the sum of variable S1 and variable S2. S1 and S2 should be input as floating-point values.
Return Value	Returns the result (floating-point value).
Example	$W0=FADD($W10,$W20); 'Set the sum of $W10(floating-point value(2 words)) and $W20(floating-point value(2 words)) to $W0(floating-point value(2 words)). $W100=FADD(2.524563,-1.255669); 'Set the sum of 2.524563 and -1.255669 to $W100(floating-point value(2 words)).

FCOMP

Format	FCOMP(S1,S2)
Function	Returns the comparison result of variable S1 with variable S2. S1 and S2 should be input as floating-point values.
Return Value	Returns the result (WORD value). (ex. 0: S1=S2, -1: S1<S2, 1: S1>S2)
Example	$W0=FCOMP($W10,$W20); 'Set the comparison result of $W10(floating-point value(2 words)) with $W20(floating-point value(2 words)) to $W0(WORD value(1 word)). $W100=FCOMP(2.524563,-1.255669); 'Set the comparison result of 2.524563 with -1.255669 to $W100(WORD value(1 word)).

FDIV

Format	FDIV(S1,S2)
Function	Returns the quotient of variable S1 and variable S2. S1 and S2 should be input as floating-point values.
Return Value	Returns the result (floating-point value).
Example	$W0=FDIV($W10,$W20); 'Set the result of $W10(floating-point value(2 words)) divided by $W20(floating-point value(2 words)) to $W0(floating-point value(2 words)). $W100=FDIV(2.524563,-1.255669); 'Set the result of 2.524563 divided by -1.255669 to $W100(floating-point value(2 words)).

FLOOR

Format	FLOOR(S)
Function	Returns the value of variable S rounded DOWNWARDS to the nearest. S should be input as a floating-point value.
Return Value	Returns the result (floating-point value).
Example	$W0=FLOOR($W10); 'Set the result of FLOOR $W10(floating-point value(2 words)) to $W0(floating-point value(2 words)). $W100=FLOOR(-0.4554356); 'Set the result of FLOOR -0.4554356 to $W100(floating-point value(2 words)).

FMUL

Format	FMUL(S1,S2)
Function	Returns the product of variable S1 and variable S2. S1 and S2 should be input as floating-point values.
Return Value	Returns the result (floating-point value).
Example	$W0=FMUL($W10,$W20); 'Set the result of $W10(floating-point value(2 words)) multiplied by $W20(floating-point value(2 words)) to $W0(floating-point value(2 words)). $W100=FMUL(2.524563,-1.255669); 'Set the result of 2.524563 multiplied by -1.255669 to $W100(floating-point value(2 words)).

FOR, NEXT

Format	FOR(n), NEXT n; a LOOP
Function	n: number of interation*1 A series of statements in a computer program that are to be executed repeatedly at specified times. A program between "FOR to NEXT" cannot be nested in another "FOR to NEXT". (Single loop only) $W, $HW and $SW can specified as an address.
Return Value	None
Example	'Execute a loop "FOR to NEXT" 10 times and substitute $W0 to '$W9 to the initial value,0. $W0=0; $SW27=0; FOR(10) $W0I0=0; $SW27=$SW27+1; NEXT;

*1. Setting range is 0 to 32767. A negative number is considered as 0.

FSET

Format	FSET(S)
Function	Returns the variable S converted to floating-point.
Return Value	Returns the result (floating-point value).
Example	$W0=FSET($W10); 'Set $W10(floating-point value(2 words)) to $W0(floating-point value(2 words)). $W100=FSET(2.524563); 'Set 2.524563 to $W100(floating-point value(2 words)).

FSUB

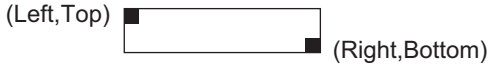
Format	FSUB(S1,S2)
Function	Returns the difference of variable S1 and variable S2. S1 and S2 should be input as floating-point values.
Return Value	Returns the result (floating-point value).
Example	$W0=FSUB($W10,$W20); 'Set the difference of $W10(floating-point value(2 words)) and $W20(floating-point value(2 words)) to $W0(floating-point value(2 words)). $W100=FSUB(2.524563,-1.255669); 'Set the difference of 2.524563 and -1.255669 to $W100(floating-point value(2 words)).

GETNUMVAL

Outputs written value and changed value

Format	GETNUMVAL()
Function	Get writing numeral value or changing numeral value for numeral display & input object. This function can be used when the execution conditions for the macro of numerical display & input object are the following: "Before Inputting Numeral", "Before writing numeral", "When changing numeral"
Return Value	Input numeral value
Example	$W0=GETNUMVAL(); 'Set value for writing numeral value to $W0

GETPARTS

Format	GETPARTS(n,Left,Top,Right,Bottom)
Function	Get displayed rectangle of the object ID number "n". Set coordinate (Left, Top) at the upper left, (Right, Bottom) at the lower right on rectangle.  Setting range is 0 to 1023. If you set other value or ID number that does not exist, return value 1 will be returned.
Return Value	0: Completed normally -1: Specified no object
Example	GETPARTS(1,$W0,$W1,$W2,$W3); 'Set coordinate of displayed rectangle of object ID number1 'to ($W0,$W1)-($W2,$W3)

LOCALTIME

Format	LOCALTIME(S,D)
Function	The local time stored in S (in nanoseconds)*1 is converted to a calendar time (year, month, day, hour, minutes, and seconds) and output to D. The output calendar time will be the UTC.
Return Value	0: Normal end -1: An out-of-range value was set for S or D. -2: A value that exceeds the upper limit was set for the local time.
Example	READCMEM($W0,[DATEANDTIME],4); $W100=LOCALTIME($W0,$W10); SETTIME($W10); 'The value of DATEANDTIME variable is converted to a calendar time and set in $W10.

*1. The local time that is stored in S cannot exceed 03:14.07 on January 9, 2062.

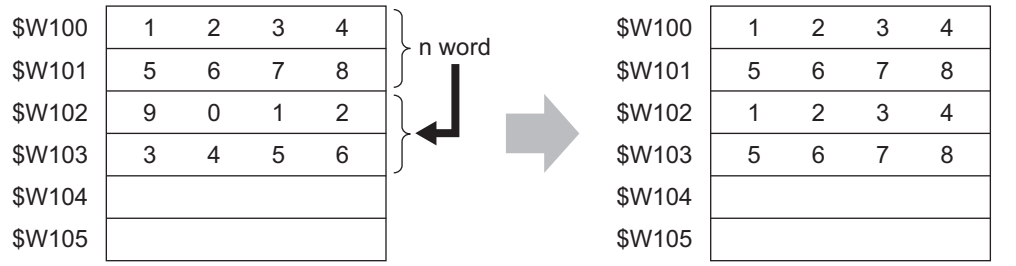
LOG

Format	LOG(S)
Function	Returns the natural logarithm of variable S. S should be input as a floating-point value.
Return Value	Returns the result (floating-point value).
Example	$W0=LOG($W10); 'Set the result of LOG $W10(floating-point value(2 words)) to $W0(floating-point value(2 words)). $W100=LOG(6.852943); 'Set the result of LOG 6.852943 to $W100(floating-point value(2 words)).

LOG10

Format	LOG10(S)
Function	Returns the base-10 logarithm of variable S. S should be input as a floating-point value.
Return Value	Returns the result (floating-point value).
Example	$W0=LOG10($W10); 'Set the result of LOG10 $W10(floating-point value(2 words)) to $W0(floating-point value(2 words)). $W100=LOG10(6.987123); 'Set the result of LOG10 6.987123 to $W100(floating-point value(2 words)).

MEMCOPY

Format	MEMCOPY(S,D,n)
Function	Copy data of $W or $HW in the PT memory. S: Top address of source data.*1 D: Top address to which data will be copied.*1 n: The number of word data which will be taken from S.*1 Setting range is as follows: When specifying “n” directly: 1 to 32767 When specifying “n” indirectly: $W0 to $W32767 $HW0 to $HW8191
Return Value	None
Example	MEMCOPY($W100,$W102,2); Take 2 words from $W 100 and copy to 102 <Before executing MEMCOPY> <After executing MEMCOPY> 

*1. Index can be set when using address to specify for S. Setting range is for $W is between 0 and 32767 and for $HW is between 0 to 8191.

MEMSET

Format	MEMSET(D,c,n)
Function	Writes data c for n-word from a word address D in the PT memory ($W/$HW). D: Starting address c: Set value -32767 to 32768 (decimal format) H0000 to HFFFF (hexadecimal format) n: Number of elements to write 1 to 32768 ($W) 1 to 8192 ($HW)
Return Value	None
Example	• Writing 5 for 10 words from $W100 ($W100 to $W109) MEMSET($W100,5,10); • Writes 5 for 10 words from $W100 ($W100 to $W109) $HW100=5; $W200=10; MEMSET($W100,$HW100,$W200);

MOVEPARTS

Format	MOVEPARTS(n,x,y)
Function	Move the object ID number “n” to specified coordinate (x,y). Specify coordinate upper left of the moving object for “x,y”. Setting range for “n” is 0 to 1023. If the value out side the range or ID number that does not exist is specified, return value -1 will be returned. There is no restriction on setting value for “x,y”. However, set the value for the coordinate of x and y in order that the objects are displayed inside of the screen. All objects or some objects on the screen may be deleted depending on the set value so care must be taken.
Return Value	0: Completed normally -1: Specified no object
Example	MOVEPARTS(3,150,200); ‘Move the object ID number 3 to position (150,200)



Additional Information

When setting macro “MOVEPARTS” for ON/OFF button, Word button, Command button and Multifunction Object and moving these objects or these objects with frame, select “Touch Off Timing”.

If “Touch On Timing” is selected, the status of the object will be pressed.

MOVEPOPW

Format	MOVEPOPW(n,x,y)
Function	Moves top left of the pop-up window for screen number “n” to the specified coordinates (x,y). Setting range for “n” is 0 to 3999. If the value outside of the range or screen number that does not exist is specified, return value “-1” will be returned. There is no restriction on setting the value for “x, y”. However, set the value for the coordinate of x and y in order that the objects are displayed inside of the screen. Part of the screen or whole screen may be deleted depending on the set value.
Return Value	0: Completed normally -1: Specified no page
Example	$W0=MOVEPOPW(10,140,160); ‘Moving pop-up screen page number 10 to the specified position (140,160), then ‘return “0” to $W0. If pop-up screen is not displayed, return “-1” to $W0.



Additional Information

When setting macro “MOVEPOPW” for ON/OFF button, Word button, Command button and Multifunction Object on a pop-up screen and moving the pop-up screen, select “Touch Off Timing”.

If “Touch On Timing” is selected, the status of the object will be pressed.

MOVEPOPWDOWN

Format	MOVEPOPWDOWN(n,y)
Function	Move the pop-up window page number “n” to y down. Setting range for “n” is 0 to 3999. If the value outside of the range or page number which does not existed is specified, return value “-1” will be returned. There is no restriction on setting value for “y”. However, set the value to the pop-up screen in order to be displayed inside of the screen. Part of the screen or whole screen may be deleted depending on the set value.
Return Value	0: Completed normally -1: Specified no page
Example	$W0=MOVEPOPWDOWN(10,32); ‘Move the pop-up window page number 10 to 32dot down, ‘then return “0” to $W0. If pop-up screen is not displayed, ‘return “-1” to #W0.



Additional Information

When setting macro “MOVEPOPWDOWN” for ON/OFF button, Word button, Command button and Multifunction Object on a pop-up screen and moving the pop-up screen, select “Touch Off Timing”.

If “Touch On Timing” is selected, the status of the object will be pressed.

MOVEPOPWLEFT

Format	MOVEPOPWLEFT(n,x)
Function	Move the pop-up window page number “n” to x dot left. Setting rage for “x” is 0 to 3999. If the value outside of the range or page number that does not exist is specified, return value “-1” is returned. There is no restriction on setting the value for “x”. However, set the value in order to be displayed inside of the screen. It may be deleted part of the screen or whole screen depends on the value.
Return Value	0: Completed normally -1: Specified no page
Example	$W0=MOVEPOPWLEFT(10,32); ‘Move pop-up window page number 10 to 32 dot left, then return ”0” to $W0. If ‘pop-up screen is not displayed, return “-1” to $W0.



Additional Information

When setting macro “MOVEPOPWLEFT” for ON/OFF button, Word button, Command button and Multifunction Object on a pop-up screen and moving the pop-up screen, select “Touch Off Timing”.

If “Touch On Timing” is selected, the status of the object will be pressed.

MOVEPOPWRIGHT

Format	MOVEPOPWRIGHT(n,x)
Function	Move pop-up window page “n” to x dot right. Setting range for “n” is 0 to 3999. If the value outside the range or screen number that does not exist is specified, return value “-1” is returned. There is no restriction on setting for “x”. However, set the value in order to be displayed inside of the screen. Part of the screen or whole screen may be deleted depending on the set value.
Return Value	0: Completed normally -1: Specified no page
Example	$W0=MOVEPOPWRIGHT(10,32); ‘Move the pop-up screen page 10 to 32 dot right, then return “0” to $W0. If the ‘pop-up screen is not displayed, return “-1” to $W0.



Additional Information

When setting macro “MOVEPOPWRIGHT” for ON/OFF button, Word button, Command button and Multifunction Object on a pop-up screen and moving the pop-up screen, select “Touch Off Timing”.

If “Touch On Timing” is selected, the status of the object will be pressed.

MOVEPOPWUP

Format	MOVEPOPWUP(n,y)
Function	Move the pop-up screen page “n” to y dot up. Setting range for “n” is 0 to 3999. If the value outside of the range or page number that does not exist is specified, return value “-1” is returned. There is no restriction on setting the value for “y”. However, set the value in order to be displayed inside of the screen. Part of the screen or whole screen may be deleted depending on the set value.
Return Value	0: Completed normally -1: Specified no page
Example	$W0=MOVEPOPWUP(10,32); ‘Move pop-up screen page 10 to 32 dot up, then return “0” to $W0. If pop-up ‘screen is not displayed, return “-1” to $W0.

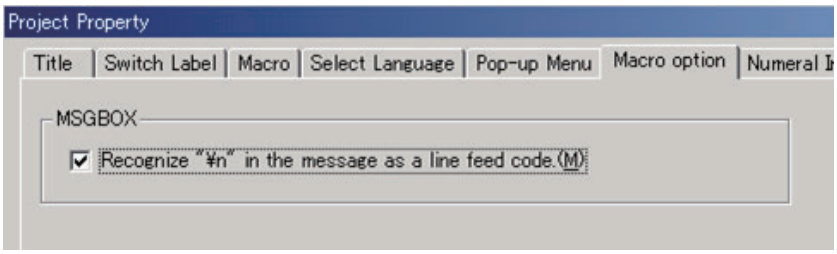


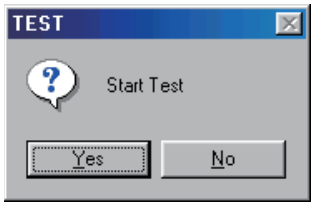
Additional Information

When setting macro “MOVEPOPWUP” for ON/OFF button, Word button, Command button and Multifunction Object on a pop-up screen and moving the pop-up screen, select “Touch Off Timing”.

If “Touch On Timing” is selected, the status of the object will be pressed.

MSGBOX

Format	MSGBOX(S1,S2,S3)
Function	Display message dialog which is specified. S1: message string Perform the following procedure to insert a line feed in the message. Select *PT - Project Properties* in the CX-Designer to display the Project Property Dialog box. Checking “Recognize “\n” in the message as a line feed code” of the MSGBOX option in the Macro option tab enables to insert a line feed by typing “\n” in the message string.  S2: title bar string S3: reply with icon type that is displayed in message dialog Specify type of the button. • 4 bits (B0-B3) 0: STOP Mark 1: QUESTION Mark 2: EXCLAMATION Mark 3: INFORMATION Mark • 4 bits (B4-B7) 0: [OK] button only 1: [OK]/[Cancel] button 2: [Retry]/[Cancel] button 3: [Yes]/[No] button 4: [Yes]/[No]/[Cancel] button 5: [Stop]/[Retry]/[Ignore] button
Return Value	0: Select [OK] button 1: Select [Cancel] button 2: Select [Yes] button 3: Select [No] button 4: Select [Ignore] button 5: Select [Retry] button 6: Select [Stop] button

Example	$W0=MSGBOX (“Start Test”,“TEST”,H31);
	‘H31:[Yes]/[No] button, Display QUESTION mark
	IF($W0==2)
	‘Write processing if you select “Yes”
	ELSE
	‘Write processing if you select “No”
	ENDIF
	
	As you set above, message dialog is displayed below
	

Additional Information

- Only one message box can be displayed using MSGBOX.
  If executing MSGBOX with displaying other message box, new message box is not displayed and “1” is returned as return value.
  Example: making two bit lamps

	address	macro
Lamp 1	$B0	$W0=MSGBOX(“message1”, “title1”, H31);
Lamp 2	$B0	$W1=MSGBOX(“message2”, “title2”, H31);

  Suppose macro of lamp1 is executed first. When changing the value of $B0, the message box of lamp 1 is displayed. Message box of lamp 2 is not displayed and value “1” is stored in $W1.
- If Buzzer Sound is set ON or OFF at PT tab in the system menu and STOP or EXCLAMATION is specified for the icon, buzzer will be sounded when displaying the message dialog.

POW

Format	POW(S1,S2)
Function	Returns the result of calculating variable S1 to the power of variable S2. S should be input as a floating-point value.
Return Value	Returns the result (floating-point value).
Example	$W0=POW($W10,$W20); ‘Set the result of calculating POW $W10,$W20(floating-point value(2 words)) to $W0(floating-point value(2 words)). $W100=POW(4.238976,0.001411); ‘Set the result of POW, calculating 4.238976 to the power of 0.001411, to $W100(floating-point value(2 words)).

RAD2DEG

Format	RAD2DEG(S)
Function	Convert the variable S from radian to its degree and return the value. S should be input as a floating-point value.
Return Value	Returns the result of conversion (floating-point value).
Example	$W0=RAD2DEG($W10); 'Set the result of RAD2DEG $W10(floating-point value(2 words)) to $W0(floating-point value(2 words)). $W100=RAD2DEG(-0.236987); 'Set the result of RAD2DEG -0.236987 to $W100(floating-point value(2 words)).

READCF

Format	READCF(Mem,Size,File,Dev)
Function	Reads the contents (binary format) of the specified file in a removable media to PT memory. Mem: Top address of destination. ($W, $HW or $SW) Index can be set. Setting range for $W is between 0 and 32767 and for $HW is between 0 to 8191. $B, $HB and $SB cannot be specified. Size: Data size to be read. (unit: word) Data size can be specified using long word directly, or $W or $HW (uses 2 words) indirectly. Setting range for $W is 0 to 32767 and for $HW is between 0 and 8191. If the set value is exceeded the maximum, an access error occurs and an error message appears. If the value 0 or less is set for Size, the specified size will be read to the PT memory. If the file size is bigger than the value set for Size (Size>0), it executes reading data set for Size. (Return value will be 0.) If the file size is smaller than the value set for Size (Size>0), it executes reading actual file size only. (Return value will be 0.) File: Source file name The file name can be specified using character string directly, or $W or $HW (uses 2 words) indirectly. Read action is executed by word unit, however, the last 1 byte of address will not be read if file size is odd byte. (Return value will be 0). Up to 43 alphanumerical characters (“0 to 9”, “A to Z”, “ a to z”, “$”, “_”) including extension can be set for the file name. Dev: Specify destination device. Always specify 0 since destination will be a removable media only.
Return Value	0: Completed normally -1: Failed to read data
Example	• $W100=READCF($W1000,0,“CF_FILE.BIN”,0); • $W2000L=0; STRCPY($W2002,“CF_FILE.BIN”); $W100=READCF($W1000,$W2000,$W2002,0);

READCMEM

Format	READCMEM(D,S,n)
Function	Read n but/n channel from the address in the host specified with “s”, and copy to “D”. Maximum points for reading are indicated below. • Bit 126 Bit • Word 126 Channel If value outside of the range is set for “n”, communication error or macro execution error occurs. Maximum points of reading depend on PLC type.
Return Value	None
Example	READCMEM($W0,[HOST1:DM0],10); ‘Read the value “$W0 to $W9” to “DM0 to DM9” at the host ‘named Host1 in PLC.

Additional Information

When a tag is specified for S, set the value in the table for n according to the tag data type.

Tag data type	Value to be set for n
BOOL	1
SINT, USINT, BYTE, UINT, INT, UINT_BCD, CHANNEL, WORD	1
UDINT, DINT, UDINT_BCD, REAL, DWORD	2
ULINT, LINT, LINT_BCD, LREAL, LWORD, TIME, DATE, TIME_OF_DAY, DATE_AND_TIME	4
STRING, CJ_STRING	No. of characters or less*1
Array	No. of elements or less*1

*1. Up to 126 can be set.

Example: When the data type of [HOST1:TAG1] is DINT.

READCMEM($W0,[HOST1:TAG1],2);

READHOSTB

Format	READHOSTB(D,h,ch,addr,r,n)
Function	Reads n-bit data from the host, h and copies it to the PT memory ($B/$HB), D. D: Starting address to read data to ($B0 to $B32767, $HB0 to $HB8191) h: Host (host name / host number) ch: Host address type*1 addr: Starting address in the host r: Bits n: Number of elements to write (1 to 126)
Return Value	Normal termination: 0x0000 Error: high order 8 bits (B8 to B15): MRES (main response cord) low order 8 bits (B0 to B7) : SRES (sub-response cord) Refer to *NS-B-series Programmable Terminal Software User’s Manual (Cat. No. V130)* for MRES and SRES.

Example	SerialA and SerialB are registered in the host: • Reads 10-bit data from CIO1000.00 in the PLC connected to the host 1 (Serial port A) and stores it to $B10 to $B19. READHOSTB($B10,1,100,1000,0,10); • Reads 10-bit data from DM2000.05 in the PLC connected to the host name=[Serial B] (Serial port B) and stores it to $HB10 to $HB19. READHOSTB($HB10,[SerialB],300,2000,5,10);

*1. Refer to *2-3 Address Type Number* on page 2-38.



Additional Information

- If a host is deleted with CX-Designer, the subsequent hosts will be renumbered in order.

  The example below shows when HOST2 is deleted, the host numbers are renumbered and HOST3's number will be changed to 2.

Before Deleting Host		After Deleting Host	
Host Number	**Host Name**	**Host Number**	**Host Name**
1	SERIALA	1	SERIALA
2	HOST2	2	HOST3
3	HOST3	---	---

  Please note that when a host number is given as argument h of READHOSTB, it is necessary to specify the host number after change. With the above example, please change the macro as follows.

  Macro Before Deleting Host READHOSTB($B10,3,100,1000,0,10);

  ↓ Change "3", the 2nd argument that specifies HOST3, to "2"

  Macro After Deleting Host READHOSTB($B10,2,100,1000,0,10);

  To check the host number, confirm the number at the left side of host name shown on the Comm.Setting dialog of CX-Designer.

- Also, note that if a host name is changed with CX-Designer when the host name is used as argument h of READHOSTB, it is necessary to specify the host name after change.

  For example, if a host name is changed from HOST_1 to HOST_A, please change as follows.

  Macro Before Changing Host Name READHOSTB($B10,[HOST_1],100,1000,0,10);

  ↓ Change HOST_1 to HOST_A

  Macro After Changing Host Name READHOSTB($B10,[HOST_A],100,1000,0,10);

READHOSTW

Format	READHOSTW(D,h,ch,addr,n)
Function	Reads n-word data from the host (h) and copies it to the PT memory ($W/$HW), D. D: Starting address to read data to ($W0 to $W32767, $HW0 to $HW8191) h: Host (host name/host number) ch: Host address type*1 addr: Host starting address n: Number of elements to write (1 to 126)
Return Value	Normal termination: 0x0000 Error: high order 8 bits (B8 to B15): MRES (main response cord) low order 8 bits (B0 to B7): SRES (sub-response cord) Refer to *NS-B-series Programmable Terminal Software User's Manual (Cat. No. V130)* for MRES and SRES.
Example	SerialA and SerialB are registered in the host: • Reads 10-bit data from CIO1000 in the PLC connected to the host 1 (Serial port A) and stores it to $W10 to $W19. READHOSTW($W10,1,100,1000,10); • Reads 10-bit data from DM2000 in the PLC connected to the host name=[Serial B] (Serial port B) and stores it to $HW10 to $HW19. READHOSTW($HW10,[SerialB],300,2000,10);

*1. Refer to *2-3 Address Type Number* on page 2-38.

Additional Information

- If a host is deleted with CX-Designer, the subsequent hosts will be renumbered in order.

  The example below shows when HOST2 is deleted, the host numbers are renumbered and HOST3's number will be changed to 2.

Before Deleting Host		After Deleting Host	
Host Number	**Host Name**	**Host Number**	**Host Name**
1	SERIALA	1	SERIALA
2	HOST2	2	HOST3
3	HOST3	---	---

Please note that when a host number is given as argument h of READHOSTW, it is necessary to specify the host number after change. With the above example, please change the macro as follows.

Macro Before Deleting Host READHOSTW($W10,3,100,1000,10);

Change "3", the 2nd argument that specifies HOST3, to "2"

Macro After Deleting Host READHOSTW($W10,2,100,1000,10);

To check the host number, confirm the number at the left side of host name shown on the Comm. Setting dialog of CX-Designer.

- Also, note that if a host name is changed with CX-Designer when the host name is used as argument h of READHOSTW, it is necessary to specify the host name after change.

  For example, if a host name is changed from HOST_1 to HOST_A, please change as follows.

Macro Before Changing Host Name READHOSTW($W10,[HOST_1],100,1000,10);

Change HOST_1 to HOST_A

Macro After Changing Host Name READHOSTW($W10,[HOST_A],100,1000,10);

RELEASEFOCUS

Format	RELEASEFOCUS()
Function	If the input focus has been set for any of numeral display & input object or string display & input object in the project, this macro will release the input focus. • If an object that the input focus has been set exists on the screen currently displayed, the macro will release the input focus. • If the input focus is not set for any object on the screen currently displayed, this macro will NOT work. • If the input focus has been set for the object created in the frame page displayed as top, this macro will also release the focus. • If the input focus has been set for the object in the sheet, this macro will release the focus. RELEASEFOCUS macro will NOT work in the following execution timing. • When Loading a Project • When Loading a Screen • When Unloading a Screen • Before Inputting Numeral set using numeral display & input objects • Before Writing Numeral set using numeral display & input objects • Before Inputting String set using string display & input objects • Before Writing String set using string display & input objects
Return Value	None
Example	In all cases that you want to release the input focus, set as the following example. RELEASEFOCUS();

RETURN

Format	RETURN(S)
Function	If the value of “S” is “0”, terminate macro program and continue to process for functional object. If a value is set other than “0”, terminate program and stop processing for functional object.
Return Value	None
Example	RETURN(0); ‘terminate macro and continue to process RETURN(1); ‘terminate macro and stop processing

RSTALARMCNT

Format	RSTALARMCNT(S)
Function	When the value of “S” is 0, clear the number of occurrence of alarm. When the value of “s” is 1, clear the number of occurrence of event.
Return Value	0: Completed normally -1: None
Example	RSTALARMCNT(0); ‘clear the number of occurrence of alarm RSTALARMCNT(1); ‘clear the number of occurrence of event

SETFOCUS

Format	SETFOCUS(n)
Function	Set the input focus on the specified numeral display & input object or string display & input object. n: Object ID number which the input focus should be set. (0 to 32767) When setting the input focus for the object specified as top in the ***Input Order List***, set “-1”. • This macro will NOT work if the input focus has already been set for other object. • The input focus will NOT be set if the specified object is created in the frame page which is not displayed as top. (A dialog which indicates macro execution error will be displayed when executing this macro.) • If an object other than numeral display & input object and string display & input is specified, this macro will NOT work. • The input focus cannot be set for the objects created in the sheet. SETFOCUS macro will NOT work in the following execution timing. • When Loading a Project • When Loading a Screen • When Unloading a Screen • Before inputting Numeral set using numeral display & input objects • Before Writing Numeral set using numeral display & input objects • Before inputting String set using string display & input objects • Before Writing String set using string display & input objects
Return Value	0: Completed normally -1: The specified object ID could not be found.
Example	Case that the input focus is set for object with ID number 4. SETFOCUS(4);

SETTIME

Format	SETTIME(S)
Function	Preset values for the specified address as S. The function writes them to the internal clock of PT Unit. Setting range for S is between $W0 and $W32765 or between $HW0 and $HW8189.*1 Set the values in BCD or BIN format for addresses to be written.*2 The function uses 3 words regarding the specified address as top.*3
Return Value	None
Example	• To read the internal clock data of PLC (in BCD format), and set it in the internal clock of the PT Unit; 'PLC (CS-, CJ- or CP-series) clock data : Read AR351 through AR353. READCMEM($W1000,[SERIALA:AR351],3); SETTIME($W1000); • To set a date and time "2002-12-31T18:59:30" in BIN format; $W100=BIN(H30); 'Second data $W101=BIN(H59); SWAP($W101,1); 'Minute data $W102=BIN(H18); 'Hour data $W103=BIN(H31); SWAP($W103,1); 'Date data $W104=BIN(H12); 'Month data $W105=BIN(H02); SWAP($W105,1); 'Year data $W200=$W100+$W101; $W201=$W102+$W103; $W202=$W104+$W105; SETTIME($W200);

*1. Index can be used for specifying the address.

*2. See the *Additional Information* in below.

*3. The date data for 3 words must be stored as shown in the table below.

Address	Higher byte (Bit08 to 15)	Lower byte (Bit00 to 07)
1st word	Minutes (00 to 59)	Second (00 to 59)
2nd word	Date (01 to 31)	Hour (00 to 23)
3rd word	Month (01 to 12)	Year (00 to 99)

Additional Information

The data format of the argument S (Date data) must be same as the storage format of the system memory $SW14 through $SW17 (Date and time).

If they are different, the SETTIME is not executed.

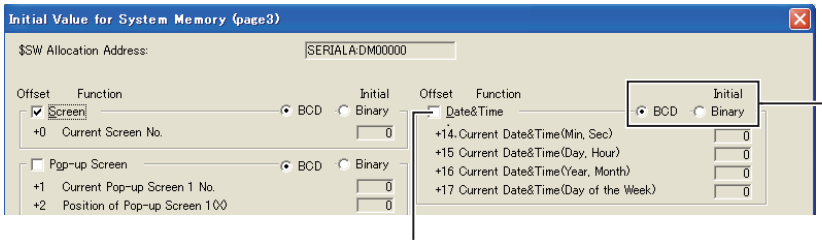
		Data format of SETTIME argument S	
		BCD	**Binary**
Storage format of $SW14 through 17	**BCD**	Yes	No
	Binary	No	Yes

Yes: SETTIME can be executed.

No: An Error Message is shown telling “An error occurred during macro execution.”

Set the same format for the SETTIME argument S and for $SW14 through 17.

To set or change the system memory storage format, select **System Setting** - **Initial** tab and press the **System Memory List** button.



* Checked or unchecked is irrelevant.

Format (BCD by fault)

SHOWPAGE

Format	SHOWPAGE(S)
Function	Switch screen to the page that is specified in “s”. Setting range for “s” is 0 to 3999. If the value outside of the range is set, macro execution error occurs. If the screen number that does not exist is set, reading page error occurs.
Return Value	None
Example	SHOWPAGE(10); ‘Switch screen to page 10



Additional Information

Macro written after SHOWPAGE is not executed. Be sure to write SHOWPAGE at the end of line.

- Bad Example:

  SHOWPAGE(10); ←Switch to page 10

  $W50=100; ←Substitute 100 to $W50 is not executed

- Good Example:

  $W50=100; ←Substitute 100 to $W50 is executed

  SHOWPAGE(10); ←Switch to page 10

Similarly, when SHOWPAGE is executed by a macro set with Multifunction object, functions set after this macro will not be executed. In order to switch screens using SHOWPAGE, set this macro as the last function to be executed.

SHOWPAGEBCD

Format	SHOWPAGEBCD(S)
Function	S: screen page number (H0 to H3999) Switch screen to the page that is specified in “S”. Setting range for “S” is H0 to H3999. If either the value outside of the range is set or an invalid value is set for BCD, macro execution error occurs. If the screen number that does not exist is set, reading page error occurs.
Return Value	None
Example	• Switch screen to page 10 SHOWPAGEBCD(H10); • Specify a screen page number indirectly to switch screen to page 10. $W100=H10; SHOWPAGEBCD($W100);



Additional Information

Macro written after SHOWPAGEBCD is not executed. Be sure to write SHOWPAGEBCD at the end of line.

- Bad Example:

  SHOWPAGEBCD(H10); ←Switch to page 10

  $W50=100; ←Substitute 100 to $W50 is not executed

- Good Example:

  $W50=100; ←Substitute 100 to $W50 is executed

  SHOWPAGEBCD(H10); ←Switch to page 10

Similarly, when SHOWPAGEBCD is executed by a macro set with Multifunction object, functions set after this macro will not be executed. In order to switch screens using SHOWPAGEBCD, set this macro as the last function to be executed.

SIN

Format	SIN(S)
Function	Returns the sine of variable S. S should be input as a floating-point value.
Return Value	Returns the sine (floating-point value). (unit: radian)
Example	$W0=SIN($W10); ‘Set the sine of $W10(floating-point value(2 words)) to $W0(2 words). $W100=SIN(-8.660942); ‘Set the sine of -8.660942 to $W100(floating-point value(2 words)).

SQRT

Format	SQRT(S)
Function	Returns the square root of variable S. S should be input as a floating-point value.
Return Value	Returns the square root (floating-point value).
Example	$W0=SQRT($W10); 'Set the square root of $W10(floating-point value(2 words)) to $W0(floating-point value(2 words)). $W100=SQRT(1.040566); 'Set the square root of 1.040566 to $W100(floating-point value(2 words)).

STRCPY(W)

Format	STRCPY(D,S)	ASCII code
	STRCPYW(D,S)	Unicode (UTF-16)
Function	Copy character string from D to S Copy is performed including null.	
Return Value	None	
Example	STRCPYW($W0,"ABC");	'Set "ABC" to $W0 to $W2
	$W100=H6400;	
	STRCPY($W110,$W100);	'Set "d" to $W110



Additional Information

- 'null' matches for "00" in ASCII code, and "0000" in Unicode (UTF-16).
- Care must be taken to set string to $W32767 because copy is performed including null. When executing STRCPY(W), string data and null may not be executed $W32767 because null is copied. (If data is exceeded $W32767, communication error occurs).

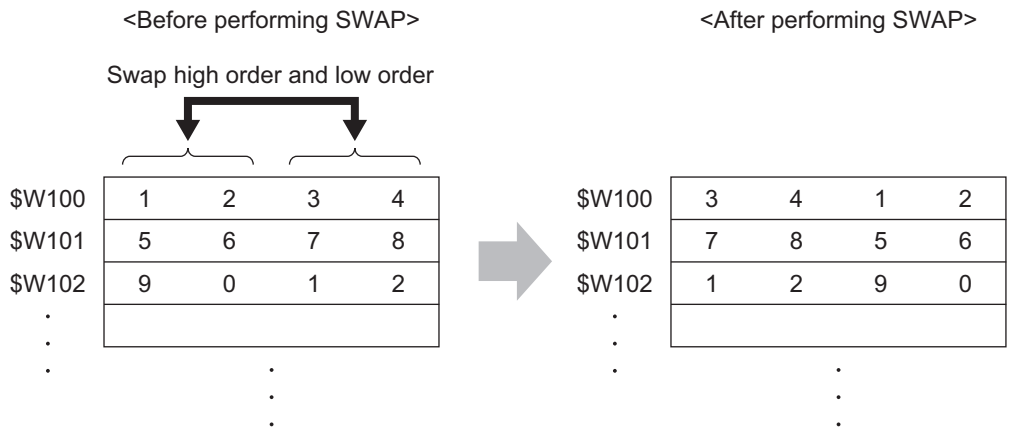
STRM2W

Format	STRM2W(D,S)
Function	Convert character string specified in "S" from ASCII code to Unicode (UTF-16) and copy to "D". Copy is performed including null.
Return Value	None
Example	STRM2W($W0,"ABC"); 'Convert "ABC" to Unicode, and copy to $W0 to $W2

STRW2M

Format	STRW2M(D,S)
Function	Convert string specified in “S” to ASCII code and copy to “D”. Copy is performed including null.
Return Value	None
Example	STRW2M($W0,“ABC”); ‘Convert “ABC” to ASCII code and copy to $W0 to $W1.

SWAP

Format	SWAP(S,n)
Function	Swap high order (1 byte) and low order (1 byte) of the word data or the internal holding word which was taken n word form S. S: Top address ($W or $HW) to be swapped.*1 n: The number of words to be taken from S.*1 Setting range is as follows: When specifying “n” directly: 1 to 32767 When specifying “n” indirectly: $W0 to $W32767 $HW0 to $HW8191
Return Value	None
Example	SWAP($W100,3); Swap high order and low order of the word data which was taken from 3 words from $W100. <Before performing SWAP> Swap high order and low order 

*1. Index can be set when using address to specify for S. Setting range is for $W is between 0 and 32767 and for $HW is between 0 to 8191.

SWAPL

Format	SWAPL(S,n)
Function	Swap high order (2 byte) and low order (2 byte) of the long word data or the internal holding word which was taken n long word from S. S: Top address ($W or $HW) to be swapped.*1 n: The number of words to be swapped.*1 Setting range is as follows: When specifying “n” directly: 1 to 16384 When specifying “n” indirectly: $W0 to $W32767 $HW0 to $HW8191
Return Value	None
Example	SWAPL($W100,3); 'Swap high order and low order of the word data which was taken from 3 long words from $W100. <Before executing SWAPL> <After executing SWAPL> Swap high order and low order of long word data

$W100	1	2	3	4		$W100	5	6	7	8
$W101	5	6	7	8		$W101	1	2	3	4
$W102	9	0	1	2		$W102	3	4	5	6
$W103	3	4	5	6		$W103	9	0	1	2
$W104	7	8	9	0		$W104	1	2	3	4
$W105	1	2	3	4		$W105	7	8	9	0

*1. Index can be set when using address to specify for S. Setting range is for $W is between 0 and 32767 and for $HW is between 0 to 8191.

TAN

Format	TAN(S)
Function	Returns the tangent of variable S. S should be input as a floating-point value.
Return Value	Returns the tangent (floating-point value). (unit: radian)
Example	$W0=TAN($W10); ‘Set the tangent of $W10(floating-point value(2 words)) to $W0(floating-point value(2 words)). $W100=TAN(-5.731245); ‘Set the tangent of -5.731245 to $W100(floating-point value(2 words)).

WRITECF

Format	WRITECF(Mem,Size,File,Dev)
Function	Saves the contents of PT memory in the specified file of removable media. The contents of the specified address will be written to the file in binary format. Mem: Top address of source data. ($W, $HW or $SW) Index can be used. Setting range for $W is between 0 and 32767, and for $HW is between 0 and 8191. $B, $HB or $SB can not be specified. Size: Data size to be saved in a removable media. (unit; word) Data size can be specified using long word directly, $W or $HW (uses 2 words) indirectly. Setting range for $W is between 0 and 32767 and for $HW is between 0 and 8191. If the set value has been exceeded the maximum, an access error will occur and an error message will appear File: Destination file name The file name can be specified using character string directly or using $W or $HW (uses 2 words) indirectly. If the specified file name for “F” already exists, the file name will be overwritten without showing a confirmation message. (Return value will be 0 (Completed normally)). Up to 43 alphanumerical characters (“0 to 9”, “A to Z”, “a to z”, “$”, “_”) including extension can be set for the file name. Dev: Always specify 0 since destination will be a removable media only.
Return Value	0: Completed normally -1: Failed to save data
Example	• $W100=WRITECF($W1000,128,“CF_FILE.BIN”,0); • $W2000L=128; STRCPY($W2002,“CF_FILE.BIN”); $W100=WRITECF($W1000,$W2000,$W2002,0);

WRITECMEM

Format	WRITECMEM(D,S,n)
Function	Copy data of n bit/n channel from “S” to the address in the host specified in “D”. Maximum points of writing are indicated below. • Bit 126 Bit • Word 126 Channel If the value outside of range is set for “n”, communication error or macro execution error occurs. Maximum points of writing depends on PLC type.
Return Value	None
Example	WRITECMEM([HOST1:DM0],$W0,10); ‘Write the value $W0 to $W9 at the host named HOST1 in PLC.

Additional Information

When a tag is specified for D, set the value in the table for n according to the tag data type.

Tag data type	Value to be set for n
BOOL	1
SINT, USINT, BYTE, UINT, INT, UINT_BCD, CHANNEL, WORD	1
UDINT, DINT, UDINT_BCD, REAL, DWORD	2
ULINT, LINTU, LINT_BCD, LREAL, LWORD, TIME, DATE, TIME_OF_DAY, DAY_AND_TIME	4
STRING, CJ_STRING	No. of characters or less*1
Array	No. of elements or less*1

*1. Up to 126 can be set.

Example: When the data type of [HOST1:TAG1] is DINT.
WRITECMEM([HOST1:TAG1],$W0,2);

WRITEHOSTB

Format	WRITEHOSTB(h,ch,addr,r,S,n)
Function	Copies n-bit data from ($B/$HB), S in the PT memory to the specified host, h. h: Host (host name/host number) ch: Host address type*1 addr: Host starting address r: Bits S: Source starting address ($B0 to $B32767, $HB0 to $HB8191) n: Number of elements to write (1 to 126)
Return Value	Normal termination: 0x0000 Error: high order 8 bits (B8 to B15): MRES (main response cord) low order 8 bits (B0 to B7): SRES (sub-response cord) Refer to *NS-B-series Programmable Terminal Software User's Manual (Cat. No. V130)* for MRES, and SRES.
Example	SerialA and SerialB are registered in the host: • Writes $B10 to $B19 to CIO1000.00 in the PLC connected to the host 1 (Serial port A). WRITEHOSTB(1,100,1000,0,$B10,10); • Writes $HB10 to $HB19 to DM1000.05 in the PLC connected to the host name=[Serial B] (Serial port B). WRITEHOSTB([SerialB],300,1000,5,$HB10,10);

*1. Refer to *2-3 Address Type Number* on page 2-38.

Additional Information

- If a host is deleted with CX-Designer, the subsequent hosts will be renumbered in order.

  The example below shows when HOST2 is deleted, the host numbers are renumbered and HOST3's number will be changed to 2.

Before Deleting Host		After Deleting Host	
Host Number	Host Name	Host Number	Host Name
1	SERIALA	1	SERIALA
2	HOST2	2	HOST3
3	HOST3	---	---

  Please note that when a host number is given as argument h of WRITEHOSTB, it is necessary to specify the host number after change. With the above example, please change the macro as follows.

  Macro Before Deleting Host WRITEHOSTB(3,100,1000,0,$B10,10);

  ↓ Change "3", the 1st argument that specifies HOST3, to "2"

  Macro After Deleting Host WRITEHOSTB(2,100,1000,0,$B10,10);

  To check the host number, confirm the number at the left side of host name shown on the Comm.Setting dialog of CX-Designer.

- Also, note that if a host name is changed with CX-Designer when the host name is used as argument h of WRITEHOSTB, it is necessary to specify the host name after change.

  For example, if a host name is changed from HOST_1 to HOST_A, change as follows.

  Macro Before Changing Host Name WRITEHOSTB([HOST_1],100,1000,0,$B10,10);

  ↓ Change HOST_1 to HOST_A

  Macro After Changing Host Name WRITEHOSTB([HOST_A],100,1000,0,$B10,10);

WRITEHOSTW

Format	WRITEHOSTW(h,ch,addr,S,n)
Function	Copies n-word data starting ($W/$HW), S in the PT memory to the specified host, h. h: Host (host name/host number) ch: Host address type*1 addr: Host starting address S: Source starting address ($W0 to $W32767, $HW0 to $HW8191) n: Number of elements to write (1 to 126)
Return Value	Normal termination: 0x0000 Error: high order 8 bits (B8 to B15): MRES (main response cord) low order 8 bits (B0 to B7): SRES (sub-response cord) Refer to *NS-B-series Programmable Terminal Software User's Manual (Cat. No. V130)* for MRES, and SRES.
Example	SerialA and SerialB are registered in the host: • Writes $W10 to $W19 to CIO1000 in the PLC connected to the host 1 (Serial port A). WRITEHOSTW(1,100,1000,$W10,10); • Writes $HW10 to $HW19 to DM1000 in the PLC connected to the host name=[Serial B] (Serial port B). WRITEHOSTW([SerialB],300,1000,$HW10,10);

*1. Refer to *2-3 Address Type Number* on page 2-38.

Additional Information

- If a host is deleted with CX-Designer, the subsequent hosts will be renumbered in order.

  The example below shows when HOST2 is deleted, the host numbers are renumbered and HOST3's number will be changed to 2.

Before Deleting Host		After Deleting Host	
Host Number	**Host Name**	**Host Number**	**Host Name**
1	SERIALA	1	SERIALA
2	HOST2	2	HOST3
3	HOST3	---	---

Please note that when a host number is given as argument h of WRITEHOSTW, it is necessary to specify the host number after change. With the above example, please change the macro as follows.

Macro Before Deleting Host WRITEHOSTW(3,100,1000,$W10,10);

Change "3", the 1st argument that specifies HOST3, to "2"

Macro After Deleting Host WRITEHOSTW(2,100,1000,$W10,10);

To check the host number, confirm the number at the left side of host name shown on the Comm.Setting dialog of CX-Designer.

- Also, note that if a host name is changed with CX-Designer when the host name is used as argument h of WRITEHOSTW, it is necessary to specify the host name after change.
  For example, if a host name is changed from HOST_1 to HOST_A, change as follows.

Macro Before Changing Host Name WRITEHOSTW([HOST_1],100,1000,$W10,10);

Change HOST_1 to HOST_A

Macro After Changing Host Name WRITEHOSTW([HOST_A],100,1000,$W10,10);

2-3 Address Type Number

Address Type Name	Address Type Number	
	BCD	Binary
PT memory - $B	0	0
PT memory - $W	1	1
PT memory - $SB	2	2
PT memory - $SW	3	3
PT memory - $HB	4	4
PT memory - $HW	5	5
Data area (CIO)	100	64
Holding area (HR)	101	65
Auxiliary area (AR)	102	66
Link area (LR)	103	67
Work area (WR)	104	68
Timer (TIM)*1	200	C8
Counter (CNT)*1	201	C9
Data memory area (DM)	300	12C
Expansion data memory (EM)	301	12D
Expansion data memory 0 (EM0)	302	12E
Expansion data memory 1 (EM1)	303	12F
Expansion data memory 2 (EM2)	304	130
Expansion data memory 3 (EM3)	305	131
Expansion data memory 4 (EM4)	306	132
Expansion data memory 5 (EM5)	307	133
Expansion data memory 6 (EM6)	308	134
Expansion data memory 7 (EM7)	309	135
Expansion data memory 8 (EM8)	310	136
Expansion data memory 9 (EM9)	311	137
Expansion data memory A (EMA)	312	138
Expansion data memory B (EMB)	313	139
Expansion data memory C (EMC)	314	13A
Expansion data memory D (EMD)	315	13B
Expansion data memory E (EME)	316	13C
Expansion data memory F (EMF)	317	13D
Expansion data memory 10 (EM10)	318	13E
Expansion data memory 11 (EM11)	319	13F
Expansion data memory 12 (EM12)	320	140
Expansion data memory 13 (EM13)	321	141
Expansion data memory 14 (EM14)	322	142
Expansion data memory 15 (EM15)	323	143
Expansion data memory 16 (EM16)	324	144
Expansion data memory 17 (EM17)	325	145
Expansion data memory 18 (EM18)	326	146

*1. Only available for READHOSTW / WRITEHOSTW

3

Error Message List

This section describes error message which is displayed in error list box when macro is added to the project, the screen and the functional objects.

3-1 Error Message List 3-2

3-1 Error Message List

Error messages are displayed in the error list box after checking the error as shown below.

Error Message	Details	Example
Format error	The program contains grammatical error except variable name, function name, or programming terms. Check whether the input function is correct.	$W0=ABC+100;
Variable name error	Variable name is incorrect	$B0:3=1;
(is missing	The ((left parentheses symbol) is missing from a function or sentence	IF$W0==1)
No. of ( ) does not agree	The number of ( ) (parentheses) in the program does not agree	IF($W0=1)!($W1=0
Position of , is incorrect	The position of the , (comma) is incorrect	IF($W0==1),($W1==0)
Function argument error	The program contains an incorrect function argument, such as word memory being set in a position that permits bit memory only. Refer to *2-1 Table of Function and Argument* on page 2-2 and check the argument.	$W0=BCD($B0);
= Command error	The program contains an incorrect substitution statement, such as 3=10, $B0=3	$W0=”ABCDE”
End of program is incomplete	The program that was input is incomplete	$W10=10+;

If sentence error	The program contains an incorrect IF,ELSE or ENDIF is statement	IF($W0==1)!($W1==0)
		$W10=1;
		ELSE
		$W10=10;

,or; is missing	The number of ,(comma) that divides the argument is insufficient. The program is not divided by a ;(semicolon).	$W10=1
FOR Statement is mismatch	FOR statement is not closed by NEXT	FOR(3) $W0=$W0+1;
	BREAK or CONTINUE is placed outside of FOR loops.	FOR(3) $W0=$W0+1; NEXT; BREAK;

Nest of FOR statement is exceeded the max. (Single loop only)	FOR is nested too deep. Nesting of loops is not supported.	FOR(3)
		$W0=$W0+1;
		FOR(5)
		$W10=$W10+10;
		NEXT;
		NEXT;

I

Index

Index

A

ACOS 2-5
ASIN 2-5
ATAN 2-5
ATAN2 2-6

B

Basic Operational Statements 1-9
BCD 2-6
BIN 2-6
BITSET 2-7
Branches 1-8
BREAK 2-7

C

CEIL 2-7
CLOSEPOPW 2-8
Comment 1-6
Conditional Expressions 1-8
Constant 1-8
CONTINUE 2-8
COS 2-8

D

DEG2RAD 2-9
Delimiter of the Program 1-6

E

EXP 2-9

F

FADD 2-9
FCOMP 2-10
FDIV 2-10
FLOOR 2-10
FMUL 2-11
FOR, NEXT 2-11
FSET 2-11
FSUB 2-12
Functions 1-10

G

GETNUMVAL 2-12
GETPARTS 2-12

L

LOCALTIME 2-13
LOG 2-13
LOG10 2-13

M

MEMCOPY 2-14
MEMSET 2-14
MOVEPARTS 2-15
MOVEPOPW 2-15
MOVEPOPWDOWN 2-16
MOVEPOPWLEFT 2-16
MOVEPOPWRIGHT 2-17
MOVEPOPWUP 2-17
MSGBOX 2-18

P

POW 2-19

Q

Qualifier of Variable 1-7

R

RAD2DEG 2-20
READCF 2-20
READCMEM 2-21
READHOSTB 2-21
READHOSTW 2-23
RELEASEFOCUS 2-24
RETURN 2-24
RSTALARMCNT 2-25

S

SETFOCUS 2-25
SETTIME 2-26
SHOWPAGE 2-28
SHOWPAGEBCD 2-29
SIN 2-29
SQRT 2-30
STRCPY(W) 2-30
STRM2W 2-30
STRW2M 2-31
SWAP 2-31
SWAPL 2-32

T

TAN 2-32

V

Variable 1-7

W

WRITECF 2-33
WRITECMEM 2-33
WRITEHOSTB 2-34
WRITEHOSTW 2-36

OMRON Corporation **Industrial Automation Company**

Kyoto, JAPAN **Contact : www.ia.omron.com**

Regional Headquarters

OMRON EUROPE B.V.
Wegalaan 67-69, 2132 JD Hoofddorp
The Netherlands
Tel: (31) 2356-81-300 Fax: (31) 2356-81-388

OMRON ELECTRONICS LLC
2895 Greenspoint Parkway, Suite 200
Hoffman Estates, IL 60169 U.S.A.
Tel: (1) 847-843-7900 Fax: (1) 847-843-7787

OMRON ASIA PACIFIC PTE. LTD.
438B Alexandra Road, #08-01/02 Alexandra
Technopark, Singapore 119968
Tel: (65) 6835-3011 Fax: (65) 6835-3011

OMRON (CHINA) CO., LTD.
Room 2211, Bank of China Tower,
200 Yin Cheng Zhong Road,
PuDong New Area, Shanghai, 200120, China
Tel: (86) 21-6023-0333 Fax: (86) 21-5037-2388

Authorized Distributor:

©OMRON Corporation 2026 All Rights Reserved.
In the interest of product improvement,
specifications are subject to change without notice.

Cat. No. V132-E1-01 1026