编号：V075-CN5-06

NS-系列
NS12-TS00□-V1/-V2, NS12-TS01□-V1/-V2
NS10-TV00□-V1/-V2, NS10-TV01□-V1/-V2
NS8-TV00□-V1/-V2, NS8-TV01□-V1/-V2
NS8-TV10□-V1, NS8-TV11□-V1
NS5-SQ00□-V1/-V2, NS5-SQ01□-V1/-V2,
NS5-TQ00□-V2, NS5-TQ01□-V2, NS5-MQ00□-V2,
NS5-MQ01□-V2

可编程终端

宏 功 能
参 考 手 册

OMRON

OMRON

特约经销商

编号：V075-CN5-06

注：规格等随时可能变化，恕不另行通知。

上海印刷
200603S01

NS系列宏功能参考手册

2006 年 3 月修订

目录

第 1 节 宏功能概要

本节叙述使用宏的执行条件和编程步骤。

1－1 什么是宏?

宏是可以由用户初始程序执行的功能。用户可以添加 NS－Designer 中标准功能不支持的功能，例如：算术运算和条件区别。该功能允许 PT 处理屏幕显示或数据，这在以前是由 PLC 执行。它也可以降低 PLC 的负荷。在该手册中，用于执行宏的触发被称为“宏执行条件”。宏可以大致划分为下面三类执行条件。

－用于项目的执行条件
－用于屏幕的执行条件
－用于功能对象的执行条件

用于1个项目/1个屏幕的宏的数量没有限制。
一个宏可以使用多达3000个字符。换行按两个字符计算。对于行的数量也没有限制。

例子：
’Number of inputting characters11个字符＋换行(2个字符)　（注释行）
$W0＝10;　　7个字符＋换行 （2个字符）
STRCPY（$W10”,ABCDE”）;　　21个字符
在该情况下，使用了43个字符。

1－2 宏执行条件

可以为每个项目、屏幕和功能对象创建宏。也可以为下面的执行条件创建宏。

用于项目的执行条件

为项目所设置的宏执行条件如下面所示。
选择NS－Designer 中的［Settings］－［Project properties］－［Macro］选项卡，然后设置执行条件并记录宏。
关于对宏进行注册的详细资料，参考NS－Designer使用手册“6－1对宏进行注册”－“6－1－1把宏注册到项目”。

执行条件	说明
When Loading a Project	在刚好启动NS－硬件之后加载第一个屏幕之前执行
Alarm/Event ON Timing	当发生警告时执行
Alarm/Event OFF Timing	当取消警告时执行

用于屏幕的执行条件

为每个屏幕所设置的宏执行条件如下面所示。
选择NS－Designer 中的［Settings］－［Screen properties］－［Macro］选项卡，然后设置执行条件并记录宏。
关于对宏进行注册的详细资料，参考NS－Designer使用手册“6－1对宏进行注册”－“6－1－1把宏注册到项目”。

执行条件	说明
When Loading a Screen	在读出显示下一个屏幕的屏幕数据之后立即执行
When Unloading a Screen	在关闭当前屏幕之后立即执行

参考

如下所示，对宏进行操作。

	当加载一个屏幕时（在目标屏幕执行）	当卸载一个屏幕时（在目标屏幕执行）
用户画面→用户画面	执行	执行
用户画面→传送画面	不执行	执行
用户画面→系统菜单	不执行	执行
系统菜单→用户画面	执行	不执行
用户画面→屏幕保护（屏保）	不执行	不执行
屏幕保护→用户画面	不执行	不执行

用于功能对象的执行条件

可以为每个功能对象设置宏执行条件，如下面的表格中的叙述。
打开每个功能对象的属性对话框，选择［Macro］选项卡页面，然后设置执行条件并创建宏。
关于对宏进行注册的详细资料，参考NS－Designer使用手册“6－1宏注册”－“为功能对象注册”。

执行条件	说明
Touch on Timing	当按功能对象时执行。
Touch off Timing	当释放功能对象时执行。
Before Inputting numeral or character string	刚好在显示用于输入数值或字符串的0~9 数字键盘或虚拟键盘之前执行。
Before Writing numeral or character string	刚好在把数字和字符串通知主机之前执行。
When changing numeral and character string	更改地址值时执行。
When Processing Display Area	当按警告显示的显示区域时执行。
When Selecting an Alarm/Event	在显示报警/事件概述上每次选择警告/事件后执行。
When selecting a list	在显示列表选择上选择一个列表后执行。

注意

如果为功能对象设置密码，在输入密码后执行下面的宏。如果取消密码输入，将不执行宏。

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

可以为功能对象选择下面的条件。

功能对象	触摸 On 时	触摸 Off 时	当更改数字/字符串时	在输入数字/字符串之前	在写入数字/字符串之前	列表选择
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	-	-	-	-	-	-
Broken-line Graph	-	-	-	-	-	-
Bitmap	-	-	-	-	-	-
Analogue Meter	-	-	-	-	-	-
Video Display	-	-	-	-	-	-
Date	-	-	-	-	-	-
Time	-	-	-	-	-	-
Data Log Graph	-	-	-	-	-	-
Data Block Table	-	-	-	OK	OK	-

[警告/事件 对象]

功能对象	当按显示区域时	当选择报警/事件时
Alarm/Event Display	OK	-
Alarm/Event Summary	-	OK

1—3 宏编程

本节叙述宏创建步骤和编程术语。

写入一个宏的方法

程序分隔符

在每个程序的末尾放一个分号（；）作为分隔符。但对于IF（）、ELSEIF（）、ELSE（）、ENDIF则不需要。

例子：

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

注释

当为每个程序添加注释时，在句子的开始放单引号。从单引号（‘）到字符串结束将被视为一个注释。

例子：

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

写入编程术语

大写和小写都可以用于对宏进行编程，因为不区分大小写。

例子：

-MovePopwDown ( )和 MOVEPOPWDOWN( )被视为相同的功能。

-[Host1:DM0] 和[host1:dm0] 被视为相同的主机地址。

编程术语

本节叙述在本功能中使用的编程术语。

变量

可以在宏程序中使用下面的变量。

项目	说明
主机地址	当访问主机中的地址时使用功能（READCMEM和WRITECMEM）进行通信。 把地址写入［ ］中。 例子： READCMEM($W100,[HOST1:DM00000],100); ‘Read HOST1:DM00000 to DM00099 to $W100 to $W199
PT 存储器	位 内部存储器：$B　$B0 到 $B32767（每1点1位） 内部保持存储器：$HB　$HB0 到 $HB8191（每1点1位） 系统存储器：$SB　$SB0到$SB 47（每1点1位） 字 内部存储器：$W　$W 到 $W32767（每1点16位） 内部保持存储器：$HW　$HW0 到 $HW 8191（每1点16位） 系统存储器：$SW　$SW0 到 $SW36（每1点16位） 例子： $W100＝$W0＋1；‘设置数值$W0和1相加，赋给$W100
索引	索引用于处理PT存储器中的位和字。 把索引添加到地址的末尾，它将按照［指定的地址 ＋ 索引值］进行处理。 有10 个索引点（I0 到 I9）。 把 I0 到 I9 设置为 $SW27 到 $SW36 的值。 例子： $SW27 = H20; $W0I0 ＝ 123；‘$W0I0 视为 $W20 $W0加上20 ‘$W20 = 123

变量限定符

必须使用下面给出的设置变量的限定符。

当执行32位数据处理和对位进行数字处理时使用限定符。

项目	说明
双字访问（32位）	在变量的末端放置“L”。使用2个字。 $W0L＝1000000；‘把$W0、$W1作为32位访问 $W100L＝1000*1000；‘把$W100、$W101作为32位访问
位的数字访问	在变量的末端放置“:n”。指定位地址（多达32位以4为单位）的值为“n”。 例外：　如果 n＝16，输入“W”。 如果 n＝32，则输入“L”。 $B0:4 ＝ 3；　‘从$B0 到 $B3 的4位设置3（0011） $B0W ＝ 12345；‘从$B0 到 ‘$B15 的16位设置 12345(0011000000111001)。

常数

宏程序和过程中可使用的常数在下表中叙述。

项目	说明
十进制常数	当使用字（16-位）时可以输入-32768到32767；当使用2个字（32-位）时可以输入-2147483648到2147483647
十六进制常数	当使用字（16-位）时可以输入H0 到 HFFFF 当使用2个字（32-位）时可以输入H0 到 HFFFFFFFF
字符串	在字符串的首尾使用“”号。 例如：“ABCDE”

分支

可以使用下面的关键字指定条件。

项目	说明

IF
ELSEIF
ELSE
ENDIF
	在IF和ELSEIF之后，把条件表达式放在圆括号（）中。
	在末尾处总是使用ENDIF。可以输入多达8组。
	对IF句子下面输入行的数量没有限制。但是，在整个宏中使用的字符的总数一定不多于3000个字符。
	例如：
	IF($W100 == 1)　　‘如果$W100 为 1
	　$W99 = 1;
	ELSEIF($W100 == 2)　　‘如果$W100 为 2
	　$W99 = 2;
	ELSE　　‘如果$W100 不同于 1 或者 2
	　$W99 = 3;
	ENDIF

条件表达式

对于在IF句子中指定条件，使用下面的条件表达式。它对所有的数据类型（字、字的双字访问、位和位的数字访问）都起作用。

项目	说明
A == B	如果A等于B，则为“TRUE”。
A > B	如果A大于B，则为“TRUE”。
A >= B	如果A大于或等于B，则为“TRUE”。
A < B	如果A小于B，则为“TRUE”。
A <= B	如果A小于或等于B，则为“TRUE”。
A <> B A ! = B	如果A不等于B，则为“TRUE”。
A && B A AND B	如果A和B都为真，则为“TRUE”。（“AND”）
A ‖ B A OR B	如果A或B为真，则为“TRUE”。（“OR”）

参考资料

（A&&B，A>B）的结果可以赋值给变量。

例如：$B100＝$W0>100；

如果$W 的数值为“100”或更小，则$B100＝0。如果$W 的值大于“100”，则$B100＝1 。

基本操作语句

在程序中可以使用下面的操作语句。

项目	符号	例子	含义
置换	=	A = B	把B 的值赋给A
加法	+	C = A+B	把 A＋B给 C的值
减法	-	C = A–B	把 A－B给 C的值
乘法	*	C = A*B	把 A×B给 C的值
除法	/	C = A/B	把 A/B给 C的值
求余	%	C = A%B	把 A%B给 C的余数
或	|	C = A|B	对A和B进行逻辑“或”运算
与	&	C = A & B	对A和B进行逻辑“与”运算
非	!	C = !A	设置C为A的非
异或	^	C = A^B	结果C为A和B的逻辑“异或”运算的值
1的补码	~	B = ~A	把A的1－补码设置给B
移位（向左）	<<	C = A<<B	把A向左移动B位的值设置给C
移位（向右）	>>	C = A>>B	把A向右移动B位的值设置给C

> ***参考资料***
> 如果执行逻辑运算，必须在同样数据类型之间（在字、位或双字访问之间）执行运算操作。
> 例如：$SW0L=$SW10L&$W20L;’ 全部使用双字访问’

多个运算可以组合。

例如： A=(B+C)*(D+E/2)

优先次序如下：

项目	符号
高	( )
↑	~
↑	*, /,%
↑	+, -
↑	<< ,>>
↑	&
↑	^
↑	|
低	=

功能

为NS系列的宏提供下面的功能。

作用		功能
在 BCD 和 BIN 之间转换	数值（BIN 码）—> BCD 码	BCD
	BCD 码 —> 数值（BIN 码）	BIN
操作字符串	复制字符串	STRCPY/STRCPYW
	从 ASCII 码转换成 Unicode	STRM2W
	从 Unicode 转换成 ASCII 码	STRW2M
警告/事件摘要	清除警告/事件发生的数量	RSTALARMCNT
HMI 专用语句	输出写入值和更改值	GETNUMVAL
	开关屏幕	SHOWPAGE/SHOWPAGEBCD
	对象显示区域的移动	MOVEPARTS
	显示信息对话框	MSGBOX
	获取显示的对象的矩形	GETPARTS
	移动弹出窗口	MOVEPOPW
	向上移动弹出窗口	MOVEPOPWUP
	向下移动弹出窗口	MOVEPOPWDOWN
	向左移动弹出窗口	MOVEPOPWLEFT
	向右移动弹出窗口	MOVEPOPWRIGHT
	关闭弹出窗口	CLOSEPOPW
通信	从指定地址读出数据	READCMEM
	把数据写入指定地址	WRITECMEM
	从指定地址读出位数据	READHOSTB
	从指定地址读出字数据	READHOSTW
	把位数据写入指定地址	WRITEHOSTB
	把字数据写入指定地址	WRITEHOSTW
处理终止	终止宏程序	RETURN
设置日期/时间	更改 PT 的内部时钟的设置	SETTIME
读出/写入数据	把一个存储卡（CF）中指定文件的内容（二进制数值）读到 PT 存储器中	READCF
	把 PT 存储器中的内容保存到一个存储卡（CF）中	WRITECF
写到多个地址	写（0/1）到 PT 存储器中的多个位地址	BITSET
	把一个数值写到 PT 存储器中的多个字地址	MEMSET
数据操作/转换	交换指定字地址的高位（1 字节）和低位（1 字节）	SWAP

	交换指定的字数据的高位（2 字节）和低位（2 字节）	SWAPL
	复制 PT 存储器中$W 的内容	MEMCOPY
输入焦点控制	为指定的对象设置输入焦点	SETFOCUS
	释放为对象设置的焦点	RELEASEFOCUS
程序循环	程序循环	FOR, NEXT
	从程序循环中退出	BREAK
	返回到 FOR 循环的顶部	CONTINUE

当可以执行一个如下所示的功能时进行触发。

.

	项目宏			屏幕宏		功能对象宏							
	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
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
CLOSEPOPW		OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
GETNUMVAL								*		*			
GETPARTS				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
READCMEM	OK	OK	OK	OK	OK	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
SHOWPAGE	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
SHOWPAGEBCD	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
WRITECMEM	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
SETTIME	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
WRITECF	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
SWAP	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
SWAPL	OK												
SETFOCUS		OK	OK			OK	OK	OK			OK	OK	OK
RELEASEFOCUS		OK	OK			OK	OK	OK			OK	OK	OK
FOR, NEXT	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
CONTINUE	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
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
WRITEHOSTB	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

WRITE HOSTW	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
MEMSET	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

*仅数字显示和输入

增加的功能

操作	功能	注释
设置日期/时间	SETTIME	在NS－Designer 3.0版中增加
读出/写入数据	READCF	在NS－Designer 4.0版中增加（NS 系列硬件使用– V1后缀）
	WRITECF	
数据处理/转换	SWAP	
	SWAPL	
	MEMCOPY	
输入聚焦控制	SETFOCUS	在NS－Designer 5.0版中增加
	RELEASEFOCUS	
切换屏幕（BCD）	SHOWPAGEBCD	在NS－Designer 6.0版中增加
重复程序	FOR, NEXT	
	BREAK	
	CONTINUE	
通信	READHOSTB	在NS－Designer 6.2版中增加
	READHOSTW	
	WRITEHOSTB	
	WRITEHOSTW	
写入多个地址	BITSET	
	MEMSET	

第2节 功能

本节叙述何使用标准功能。

2－1 功能和参数表

变量和数值，可以被指定为一个宏功能的参数，叙述如下。下表的行中的字母，例如：S,D,n,x,y 是指在“2-2 功能详细资料”－“格式”中使用的参数。

功能	参数	PT 存储器				常量	字符串	主机侧地址	指定索引
		$B $HB $SB	位的数字访问	$W $HW $SW	双字访问				
BCD	S		○	○	○	○			○
BIN	S		○	○	○	○			○
CLOSEPOPW	n		○	○	○	○			○
GETNUMVAL	无								
GETPARTS	N		○	○	○	○			○
	Left, Top, Right, Bottom			○	○				○
MOVEPARTS	n		○	○	○	○			○
	X		○	○	○	○			○
	Y		○	○	○	○			○
MOVEPOPW	n		○	○	○	○			○
	x		○	○	○	○			○
	y		○	○	○	○			○
MOVEPOPWDOWN	n		○	○	○	○			○
	y		○	○	○	○			○
MOVEPOPWLEFT	n		○	○	○	○			○
	x		○	○	○	○			○
MOVEPOPWRIGHT	n		○	○	○	○			○
	y		○	○	○	○			○
MOVEPOPWUP	n		○	○	○	○			○
	y		○	○	○	○			○
MSGBOX	S1			○	○		○		○
	S2			○	○		○		○
	S3		○	○	○	○			○

功能	参数	PT 存储器				常量	字符串	主机侧地址	指定索引
		$B $HB $SB	位的数字存取	$W $HW $SW	双字访问				
READCMEM	D	○		○	○				○
	S	○		○	○			○	○
	N					○			
RETURN	S		○	○	○	○			○
RSTALARMCNT	S		○	○	○	○			○
SHOWPAGE	S		○	○	○	○			○
SHOWPAGEBCD	S		○	○	○	○			○
STRCPY(W)	D			○	○				○
	S			○	○		○		○
STRM2W	D			○	○				○
	S			○	○		○		○
STRM2M	D			○	○				○
	S			○	○		○		○
WRITECMEM	D	○		○	○			○	○
	S	○		○	○				○
	n	○	○	○	○				
SETTIME	S			○					
READCF	Mem			○					○
	Size			○					
	File			○			○		
	Dev			○		○			
WRITECF	Mem			○					○
	Size			○					
	File			○			○		
	Dev			○		○			
SWAP	S			○	○				○
	n			○	○	○			
SWAPL	S			○	○				○
	n			○	○	○			
MEMCOPY	S			○	○				○
	D			○	○				○
	N			○					○
SETFOCUS	N			○		○			○
RELEASEFOCUS	None								
READHOSTB	D	○							○
	h			○		○		○	
	ch			○		○			○

	addr			○		○			○
	r			○		○			○
	n			○		○			○
READHOSTW	D			○					○
	h			○		○		○	
	ch			○		○			○
	Addr			○		○			○
	n			○		○			○
WRITEHOSTB	h			○		○		○	
	ch			○		○			○
	addr			○		○			○
	r			○		○			○
	S	○							○
	n			○		○			○
WRITEHOSTW	h			○		○		○	
	ch			○		○			○
	addr			○		○			○
	S			○					○
	n			○		○			○
BITSET	D	○							○
	c	○				○			○
	n			○		○			○
MEMSET	D			○					○
	c			○		○			○
	n			○		○			○

2－2功能详细资料

本节叙述在宏编程中所使用的标准功能的详细资料。

BCD　　把数值转换成 BCD 码

适用的版本	系统版本 2 或者更高版本
格式	BCD(S)
功能	把数值“S”转换成BCD码 转换范围为 0 到 99999999 如果指定的字符串超出范围，将发生溢出。 “0”被设置在字符串的末尾。
返回值	BCD 码
例子	$W0 ＝ 1234；　‘把数值1234 设置到 $W0 $W10 ＝ BCD（$W0）；　‘把 BCD 码（H1234）设置到$W10 $W20L = 12345678; ‘把数值 12345678 设置到$W20 ～ $W21 $W22L = BCD($W20L); ‘把 BCD 码（H12345678）设置到$W22 ～ W23

BIN　　把 BCD 码转换为数值

适用的版本	系统版本 2 或者更高版本
格式	BIN(S)
功能	把BCD码S转换成数值 转换范围为H0 到 H99999999
返回值	数值
例子	$W0 ＝ H1234；　　‘把BCD码（H1234）设置到$W0 $W10 ＝ BIN（$W0）；　‘把 1234 设置到$W10 $W20L = H334455; ‘把 BCD 码（H334455）设置到$W20～$W21BCD $W22L = BIN($W20L);　　‘把 334455 设置到$W22～$W23

BITSET 把（0/1）写入 PT 存储器中的多位地址

适用的版本	系统版本 6.2 或者更高版本
格式	BITSET(D, c, n)
功能	把数据C（0/1）写入PT存储器中地址D开始的n个位中。 D：开始地址 C：设置数值（0/1） N：要写入的元素的数量 1 到 32768 （$B） 1 到 8192 （$HB）
返回值	无
例子	- 把 1 写入从$B100 开始的 10 个位（$B100～$B109） BITSET($B100, 1, 10); - 把 1 写入从$B100 开始的 10 个位（$B100～$B109） $HB100=1; $W200=10; BITSET($B100, $HB100, $W200);

BREAK 从程序循环中退出

适用的版本	系统版本 6 或者更高版本
格式	BREAK
功能	中断“FOR 和 NEXT”之间的一个循环程序
返回值	无
例子	如果$W100I0>30 为真，则退出“FOR”循环。

```
$SW27=0;
FOR(10)
 $W100I0=$W50I0+10;
 IF($W100I0>30)
        BREAK;
 ENDIF
 $SW27=$SW27+1;
NEXT;
```

* “n”的设置范围为 0～32767。一个负数被视为 0。$W、$HW 和$SW 可以被指定为一个地址。

CLOSEPOPW 关闭弹出窗口

适用的版本	系统版本 2 或者更高版本
格式	CLOSEPOPW(n)
功能	关闭页编号为“n”的弹出式窗口屏幕。 “n”的设置范围为 0～3999。如果设置不存在的弹出式屏幕页编号，将不执行操作。
返回值	无
例子	CLOSEPOPW（15）；‘关闭弹出式屏幕页 15

CONTINUE 重复程序

适用的版本	系统版本 6 或者更高版本
格式	CONTINUE
功能	在“FOR ～ NEXT”之间的一个程序的过程中，它将返回 FOR 循环的顶端，并继续 FOR 过程。
返回值	无

例子	如果$W50I0>30 为真，循环将返回顶部，并继续下一个循环程序。
	
	$SW27=0;
	FOR(10)
	  IF($W50I0>30)
	  $SW27=$SW27+1;
	CONTINUE;
	  ENDIF
	  $W100I0=$W50I0+10;
	  $SW27=$SW27+1;
	NEXT;
	
	* “n”的设置范围为 0～32767。一个负数被视为 0。$W、$HW 和$SW 可以被指定为一个地址。

FOR(n), NEXT　　n：迭代次数

适用的版本	系统版本 6 或者更高版本
格式	FOR(n), NEXT n; 一个循环
功能	要在规定次数重复执行的一个计算机程序中的一系列语句。 “FOR 到 NEXT”之间的一个程序不能嵌套在另一个“FOR 到 NEXT”程序中。（只能单循环）
返回值	无

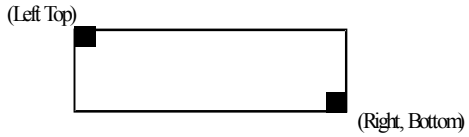
例子	执行一个循环“FOR 到 NEXT”10 次并把初始值 0 代入$W0～$W9。
	
	$W0=0;
	$SW27=0;
	FOR(10)
	  $W0I0=0;
	  $SW27=$SW27+1;
	NEXT;
	
	* “n”的设置范围为 0～32767。一个负数被视为 0。$W、$HW 和$SW 可以被指定为一个地址。

GETNUMVAL　　输出写入值和更改值

适用的版本	系统版本 2 或者更高版本
格式	GETNUMVAL( )
功能	为数字显示和输入对象获取写入的数字值或更改的数字值。 该功能用于在数字显示和输入对象中“宏执行条件”的“写入数字之前”或者“当更改数字时” 。
返回值	输入数值
例子	$W0=GETNUMVAL( ); ‘把写入数字值的值设置到$W0

GETPARTS 设定显示的矩形对象尺寸

适用的版本	系统版本 2 或者更高版本
格式	GETPARTS(n, Left, Top, Right, Bottom)
功能	获取显示的矩形对象的 ID 编号“n”。设置矩形左上角的坐标为（左，顶部），设置矩形右下角的坐标为（右，底部）。



设置范围为 0～1023。如果设置其他的数值或者不存在的 ID 编号，将返回一个返回值 1。

返回值	0：正常完成 －1：没有指定对象
例子	GETPARTS(1, $W0, $W1, $W2, $W3); ‘设置显示的矩形对象 ID 编号 1 的坐标为（$W0，$W1）－（$W2，‘$W3）

MEMCOPY 复制 PT 存储器中$W 的内容

适用的版本	系统版本 4 或者更高版本
格式	MEMCOPY (S, D, n);
功能	复制PT存储器中$W或$HW的数据。 S： 源数据的顶端地址。 D： 数据将被复制到该地址的顶端地址。 n： 将从S获取的字数据的数量。 设置范围如下： 当直接指定“n”时：1 to 32767 当间接指定“n”时：$W0 to $W32767 $HW0 to $HW8191 注释：当使用地址为 S 指定值时，可以设置索引。$W 的设置范围在 0 和 32767 之间，$HW 的设置范围在 0 到 8191 之间。
返回值	无
例子	MEMCOPY($W100, $W102, 2); 从$W 100 取两个字并复制到 102

<执行 MEMCOPY 之前>

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

n word

<执行 MEMCO

$W100	1	2
$W101	5	6
$W102	1	2
$W103	5	6
$W104		
$W105		

MEMSET 把一个数值写入 PT 存储器的多字地址中

适用的版本	系统版本 2 或者更高版本
格式	MEMSET(D, c, n)
功能	把数据c写入PT存储器（$W/$HW）中从地址D开始的n个字。 D： 起始地址 c： 设置值 -32767 to 32768 （十进制格式） H0000 to HFFFF （十六进制格式） n：要写入的元素的个数 1 to 32768 ($W) 1 to 8192 ($HW)
返回值	无
例子	- 把 5 写入从$W100 开始（$W100 ～ $W109）的 10 个字 MEMSET($W100, 5, 10); - 把 5 写入从$W100 开始（$W100 ～ $W109）的 10 个字 $HW100=5; $W200=10; MEMSET($W100, $HW100, $W200);

MOVEPARTS 移动对象显示区域

适用的版本	系统版本 2 或者更高版本
格式	MOVEPARTS (n,x,y)
功能	把ID编号“n”的对象移动到指定的坐标（x，y）。 指定移动对象的坐标左上角为“x，y”。 “n”的设置范围为 0～1023。如果数值超出范围或者指定不存在的 ID 编号，将返回一个返回值“－1”。对于“x，y”的设置值没有限制。但是，为 x 和 y 坐标设置数值，使对象显示在屏幕内。根据设置值不同，可以删除屏幕上的所有对象或某些对象，因此必须小心。
返回值	0：正常完成 －1：无指定对象
例子	MOVEPARTS (3, 150, 200); 把 ID 编号 3 的对象移动到位置（150,200）。

参考资料

当为 On/Off 按钮、字按钮、命令按钮设置宏“MOVEPARTS”，并且在移动这些按钮时请选择“触摸 Off 计时”。如果选择“触摸 On 计时”，按钮的状态将是被压紧。

使用 MOVEPARTS 不可以移动视频显示对象。

MOVEPOPW 移动弹出式窗口

适用的版本	系统版本 2 或者更高版本
格式	MOVEPOPW(n,x,y)
功能	把屏幕编号“n”的弹出式窗口的左上角移动到指定的坐标（x，y）。 “n”的设置范围为 0～3999。如果数值超出范围或者指定不存在的 ID 编号，将返回一个返回值“－1”。对于设置“x，y”的值没有限制。但是，为 x 和 y 坐标设置数值，使对象显示在屏幕内。根据设置值不同，可以删除部分屏幕或整个屏幕。
返回值	0：正常完成 －1：无指定页
例子	$W0=MOVEPOPW(10, 140, 160); ‘把页编号 10 的弹出式屏幕移动到指定的位置（140，160），然 ‘后把 0 返回给$W0。如果不显示弹出式屏幕，则把－1 返回给 ‘$W0。

MOVEPOPWDOWN 向下移动弹出式窗口

适用的版本	系统版本 2 或者更高版本
格式	MOVEPOPWDOWN(n, y)
功能	把页面编号“n”的弹出式窗口向下移动 y。“n”的设置范围为 0～3999。如果数值超出范围或者指定不存在的页面编号，将返回一个返回值“－1”。对于“y”的设置值没有限制。但是，为弹出式屏幕设置数值，使对象显示在屏幕内。根据设置值不同，可以删除部分屏幕或整个屏幕。
返回值	0：正常完成 －1：无指定页
例子	$W0=MOVEPOPWDOWN(10, 32); ‘把页编号10的弹出式屏幕向下移动32点， ‘然后把“0”返回给$W0。如果不显示弹出式屏幕，则把－1返回‘给$W0。

MOVEPOPWLEFT 向左移动弹出式窗口

适用的版本	系统版本 2 或者更高版本
格式	MOVEPOPWLEFT (n, x)
功能	把页面编号“n”的弹出式窗口向左移动 x 点。“n”的设置范围为 0～3999。如果数值超出范围或者指定不存在的页面编号，将返回一个返回值“－1”。对于设置“x”的值没有限制。但是，设置数值使对象显示在屏幕内。根据设置值不同，可以删除部分屏幕或整个屏幕。
返回值	0：正常完成 －1：无指定页
例子	$W0=MOVEPOPWLEFT (10, 32); ‘把页编号10的弹出式屏幕向左移动32点， ‘然后把“0”返回给$W0。如果不显示弹出式屏幕，则把－1 返回‘给$W0。

MOVEPOPWRIGHT 向右移动弹出式窗口

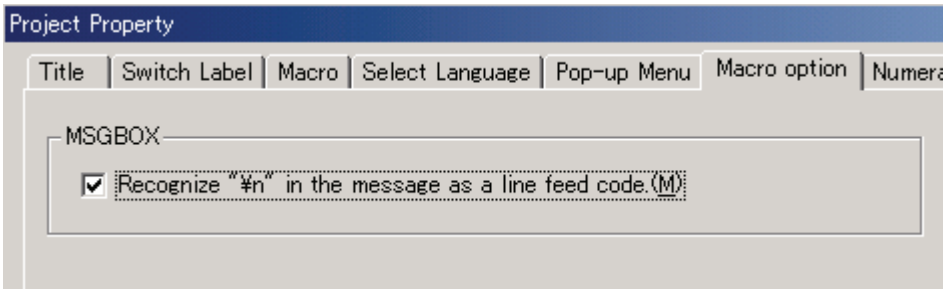
适用的版本	系统版本 2 或者更高版本
格式	MOVEPOPWRIGHT(n, x)
功能	把页面编号“n”的弹出式窗口向右移动 x 点。“n”的设置范围为 0～3999。如果数值超出范围或者指定不存在的页面编号，将返回一个返回值“－1”。对于设置“x”的值没有限制。但是，设置数值使对象显示在屏幕内。根据设置值不同，可以删除部分屏幕或整个屏幕。
返回值	0：正常完成 －1：无指定页
例子	$W0=MOVEPOPWRIGHT (10,32); ‘把页编号10的弹出式屏幕向右移动32点， ‘然后把“0”返回给$W0。如果不显示弹出式屏幕，则把－1 返回‘给$W0。

MOVEPOPWUP 向上移动弹出式窗口

适用的版本	系统版本 2 或者更高版本
格式	MOVEPOPWUP (n, y)
功能	把页面编号“n”的弹出式窗口向上移动 y。“n”的设置范围为 0～3999。如果数值超出范围或者指定不存在的页面编号，将返回一个返回值“－1”。对于“y”的设置值没有限制。但是，为弹出式屏幕设置数值，使对象显示在屏幕内。根据设置值不同，可以删除部分屏幕或整个屏幕。
返回值	0：正常完成 －1：无指定页
例子	$W0=MOVEPOPWUP (10,32); ‘把页编号10的弹出式屏幕向上移动32点，然后把“0”返回给’ ‘$W0。如果不显示弹出式屏幕，则把－1返回给$W0。

MSGBOX 显示信息对话框

适用的版本	系统版本 2 或者更高版本
格式	MSGBOX (S1, S2, S3)
功能	显示指定的信息对话框。 S1： 信息字符串 执行下面的步骤，把一个换行插入信息中。 在 NS－Designer 中选择 *Setting（设置）– Project Properties（项目属性）*来显示项目属性对话框。选中 Macro option（宏选项卡）中 MSGBOX 选项的 “。。”启用通过在信息字符串中键入“\n”来插入一个换行。



S2：标题栏字符串
S3：以信息对话框中显示的图标类型回答
指定按钮的类型。
4 位（B0－B3）

0:	停止标记	1:	问题标记
2:	执行标记	3:	信息标记

4 位（B4－B7）
0:[OK] 仅［OK］按钮　　1:[OK]/[Cancel] 按钮
2:[Retry]/[Cancel] 按钮
3:[Yes]/[No] 按钮
4:[Yes]/[No]/[Cancel] 按钮
5:[Stop]/[Retry]/[Ignore] 按钮

返回值	0： 选择［OK］按钮　　1： 选择［Cancel］按钮 2：选择［Yes］按钮　　3： 选择［No］按钮 4： 选择［Ignore］按钮　　5： 选择［Retry］按钮 6：选择［Stop］按钮

例子	$W0=MSGBOX (“Start Test”, “TEST”, H31);
	‘H31:[Yes]/[No] 按钮，显示问题标记
	IF($W0==2)
	‘如果选择“Yes”时写入操作
	ELSE
	‘如果选择“No”时写入操作
	ENDIF
	按照上面的设置，显示下面的信息对话框

参考资料

使用 MSGBOX 仅可以显示一个信息对话框。如果在显示其他信息对话框，不显示新的信息对话框时执行 MSGBOX，并返回“1”作为返回值。

例子： 制作两位灯

	地址	宏
灯 1	$B0	$W0=MSGBOX(“message1”, ”title1”, H31);
灯 2	$B0	$W1=MSGBOX(“message2”, ”title2”, H31);

假定首先执行灯 1 的宏。当更改$B0 的值时，显示灯 1 的信息对话框。不显示灯 2 的信息对话框并把数值“1”存储在$W1。

如果把系统菜单中 PT 选项卡的 Buzzer Sound（蜂鸣器声音）设置为 ON（打开）或者 OFF（关闭）并为图标指定为 STOP（停止）或者 EXCLAMATION（执行），当显示信息对话框时，蜂鸣器将发出声音。

READCF　把存储器卡（CF）中指定文件的内容（二进制）读到 PT 存储器

适用的版本	系统版本 4 或者更高版本
格式	READCF(Mem, Size, File, Dev)
功能	把存储器卡（CF）中指定文件的内容（二进制）读到PT存储器 Mem：　目的地的顶端地址。可以设置（$W，$HW 或者 $SW）索引。 $W的设置范围在0和32767之间，$HW的设置范围是在0 到8191之间。不能指定$B、$HB和$SB。 Size：　要读取的数据大小。（单位：字） 可以直接使用双字、或者间接使用$W或$HW（使用2个字）指定数据大小。$W的设置范围为0～32767，$HW的设置范围为0和8191之间。如果设置值超过最大值，将发生一个存取错误，并出现一条错误信息。 如果把Size设置为0或者小于0，将把指定大小读入PT存储器。 如果文件大小大于Size设置值（Size>0），将执行读取为Size设置的数据。 如果文件大小小于Size设置值（Size>0），仅将执行读取实际文件大小。 File：　源文件名 可以直接使用字符串、或者间接使用$W或$HW（使用2个字）指定文件名。以字为单位执行读取动作，但是，如果文件名大小为奇数字节将不读取最后一个地址字节。（返回值将为0）。可以为文件名设置多达43个字母数字混合字符（“0 ～ 9”、“.A ～ Z”、“a ～ z”、“$”，“_”），包括扩展。 Dev：　指定目的地设备。由于目的地将仅为一个存储器卡，所以，总是指定 0。
返回值	0：　正常完成 －1：　读取数据失败
例子	1. $W100=READCF($W1000,0，“ CF_FILE.BIN”,0); 2. $W2000L=0; STRCOPY($W2002, “CF_FILE.BIN”); $W100=READCF($W1000,$W2000,$W2002,0);

READCMEM 从指定的地址读取数据

适用的版本	系统版本 2 或者更高版本
格式	READCMEM (D,S,n)
功能	从用“s”指定的主机中的地址读取 n bit/n 通道，并复制到“D”。.. 读取的最大点如下面所示。 位 126 位 字 126 通道 如果为“n”设置超过范围的值，将发生通信错误或宏执行错误。读取的最大点取决于 PLC 类型。
返回值	无
例子	READCMEM ($W0, [HOST1:DM0], 10) ‘把 PLC 中主机名 Host1 中的值“DW0 ～ DW9”读到“$M0 ～ $M9”中。

READHOSTB 从指定的地址读取位数据

适用的版本	系统版本 6.2 或者更高版本
格式	READHOSTB(D, h, ch, addr, r, n)
功能	从主机h读取n位数据并把它复制到PT存储器（$B/$HB）D。 D： 读数据到（$B0 ～ $B32767，$HB0 ～ $HB8191）的起始地址 h： 主机（主机名/主机编号） ch：主机地址类型 *1 addr：主机中起始地址 r： 位 n： 要写入的元素的个数（1～126） *1：参考本章末尾的*“地址类型编号”*。
返回值	正常完成： 0x0000 出错： 高8位（B8 ～ B15）： MRES （主响应线） 低8位（B0 ～ B7） ： SRES （副响应线） * 参考用于 MRES 和 SRES 的 *5－2－7 NS－系列编程手册*中的*通信错误和措施*。
例子	把Serial A和Serial B注册到主机中： - 从与主机 1（串行口 A）相连的 PLC 中的 CIO1000.00 读取 10 位数据并把它存储到$B10 ～ $B19。 READHOSTB($B10, 1, 100, 1000, 0, 10); - 从与主机名＝［Serial B］（串行口 B）相连的 PLC 中的 CIO1000.05 读取 10 位数据并把它存储到$B10 ～ $B19。 READHOSTB($HB10, [SerialB], 300, 2000, 5, 10);

READHOSTW 从指定的地址读取字数据

适用的版本	系统版本 6.2 或者更高版本
格式	READHOSTW(D, h, ch, addr, n)
功能	从主机（h）读取n个字数据并把它复制到PT存储器（$W/$HW），D。 D： 读数据到（$W0～$W32767，$HW0～$HWB8191）的起始地址 h： 主机（主机名/主机编号） ch：主机地址类型 *1 addr：主机起始地址 n： 要写入的元素的个数（1～126） *1：参考本章末尾的*“地址类型编号”*。
返回值	正常完成： 0x0000 出错： 高8位（B8 ～ B15）： MRES （主响应线） 低8位（B0 ～ B7） ： SRES （副响应线） * 参考用于 MRES 和 SRES 的 *5－2－7 NS－系列编程手册*中的*通信错误和措施*。
例子	把Serial A和Serial B注册到主机中： - 从与主机 1（串行口 A）相连的 PLC 中的 CIO1000 读取 10 位数据并把它存储到$W10～$W19。 READHOSTW($W10, 1, 100, 1000, 10); - 从与主机名＝［Serial B］（串行口 B）相连的 PLC 中的 DM2000 读取 10 位数据并把它存储到$HW10 ～ $HW19。 READHOSTW($HW10, [SerialB], 300, 2000, 10);

RELEASEFOCUS 释放对象的输入焦点

适用的版本	系统版本 5 或者更高版本
格式	RELEASEFOCUS(); 不使用参数。
功能	如果已为项目中任何数字显示和输入对象或者字符串显示和输入对象设置输入焦点，该宏将释放输入焦点。 －如果已被设置输入焦点的一个对象存在于当前显示的屏幕上，宏将释放该输入焦点. －如果没有为当前屏幕上的任何对象设置输入焦点，该宏将不起作用。 －如果为在背景页前端显示的框架页中创建的对象设置输入焦点，该宏也将释放该焦点。 －如果已为表中的对象设置输入焦点，该宏将释放该焦点。 在下面执行触发中，RELEASEFOCUS宏将不起作用。 －当加载一个项目时（When Loading a Project） －当加载一个屏幕时（When Loading a Screen） －当卸载一个屏幕时（When Unloading a Screen） －在使用数字显示和输入对象的输入数字前（Before inputting numeral） －在使用数字显示和输入对象的写入数字之前（Before writing numeral） －在使用字符串显示和输入对象的输入字符串之前（Before inputting string） －在使用字符串显示和输入对象的写入字符串之前（Before writing string）
返回值	无
例子	在想要释放输入焦点的所有情况下，按照下面的例子进行设置。 RELEASEFOCUS();

RETURN 终止宏程序

适用的版本	系统版本 2 或者更高版本
格式	RETURN(S)
功能	如果“S”的值为“0”，终止宏程序并继续处理功能对象。如果设置不为“0”的数值，终止程序并停止对功能对象的操作。
返回值	无
例子	RETURN(0); ‘终止宏并继续处理功能对象 RETURN(1); ‘终止宏并停止处理功能对象

RSTALARMCNT 清除报警/事件发生的数量

适用的版本	系统版本 2 或者更高版本
格式	RSTALARMCNT(S)
功能	当“S”的值为 0 时，清除警告的发生次数。当“s”的值是 1 时，清除事件的发生次数。
返回值	0：正常完成 －1：无
例子	RSTALARMCNT(0); ‘清除报警的发生次数 RSTALARMCNT(1); ‘清除事件的发生次数

SETFOCUS 为对象设置输入焦点

适用的版本	系统版本 5 或者更高版本
格式	SETFOCUS(n);
功能	在指定的数字显示和输入对象或字符串显示和输入对象上设置输入焦点。 n： 用被设置输入焦点的对象ID编号。 （0～32767） 当为在***输入顺序列表***中指定作为前端的对象设置输入焦点时，设置“－1”。 －如果已经为其他对象设置输入焦点，该宏将不起作用。 －如果在不是作为前端显示的框架页中创建的指定对象，将不能设置输入焦点。（当执行该宏时，将显示一个指示宏执行出错的对话框） －如果指定一个不是数字显示和输入对象或字符串显示和输入对象的对象时，该宏将不起作用。 －不能为指定页中创建的对象设置输入焦点。 在下面执行定时中，SETFOCUS 宏将不起作用。 －当加载一个项目时（When Loading a Project） －当加载一个屏幕时（When Loading a Screen） －当卸载一个屏幕时（When Unloading a Screen） －在使用数字显示和输入对象的输入数字之前（Before inputting numberal） －在使用数字显示和输入对象的写入数字之前（Before writing numeral） －在使用字符串显示和输入对象的输入字符串之前（Before inputting string） －在使用字符串显示和输入对象的写入字符串之前（Before writing string）
返回值	0： 正常完成 －1： 找不到指定的对象 ID。
例子	为 ID 编号 4 的对象设置输入焦点的情况。 SETFOCUS(4);

SETTIME 更改 PT 内部时钟的设置

适用的版本	系统版本 3 或者更高版本
格式	SETTIME(S)
功能	为指定的地址预设置值为S并把它们写入到PT的内部时钟。S的设置值范围为$W0和$W32765之间或$HW0和$HW8189之间。（参阅注释） 设置的数值以BCD格式为写入地址。 以指定地址为起始连续使用3个字。 注释：可以使用索引指定地址。
返回值	无
例子	设置 2002 年 12 月 31 日 18：59：20 的例子如下： $W 100=H5920; $W101=H3118; $W102=H0212; SETTIME($W100);

SHOWPAGE 切换屏幕

适用的版本	系统版本 2 或者更高版本
格式	SHOWPAGE(S)
功能	把屏幕切换到“S”中指定的页面。 “S”的设置范围是 0～3999。如果设置超过范围的值，将发生宏执行错误。如果设置不存在的屏幕编号，将发生读取页面错误。
返回值	无
例子	SHOWPAGE(10); ‘把屏幕切换到页面 10

参考资料

不执行在SHOWPAGE指令后米写入的宏。一定要在行的末尾书写SHOWPAGE。
不好的例子：

```
SHOWPAGE(10);   ←切换到页面 10
$W50=100;       ←不执行把值 100 赋值给$W50
```

Good Example:

```
$W50=100;       ←执行把值 100 赋值给$W50
SHOWPAGE(10);   ←切换到页面 10
```

SHOWPAGEBCD 把屏幕切换到屏幕页 n

适用的版本	系统版本 6 或者更高版本
格式	SHOWPAGEBCD(S)
功能	S： 屏幕页编号（H0～H3999） 把屏幕切换到在“S”中指定的页面。 “S”的设置值是 H0～H3999。如果设置超出范围的值或者为 BCD 设置无效的值，将发生宏执行错误。如果设置不存在的屏幕编号，将发生读取页面错误。
返回值	无
例子	把屏幕切换到页面 10 SHOWPAGEBCD(H10); 间接地指定一个屏幕页编号，把屏幕切换到页面10： $W100＝H10； SHOWPAGEBCD($W100);

参考资料

不执行在SHOWPAGEBCD指令后面写入的宏。一定要在行的末尾书写 SHOWPAGEBCD。

不好的例子：

```
SHOWPAGEBCD(H10);   ←切换到页面 10
$W50=100;           ←不执行把值 100 赋值给$W50
```

好的例子：

```
$W50=100;           ←执行把值 100 赋值给$W50
SHOWPAGEBCD(H10);   ←切换到页面 10
```

STRCPY(W) 复制字符串

适用的版本	系统版本 2 或者更高版本
格式	STRCPY (D,S) ； ASCII 码 STRCPYW(D,S) ； Uni code
功能	从D复制字符串到S。 执行复制时包括 null（空）。
返回值	无
例子	STRCPYW ($W0, “ABC”); ‘把 “ABC” 设置到 $W0～$W2 $W100=H6400;STRCPY($W110,$W100); ‘把“d”设置到$W110

参考资料

‘null’与ASCII码中“00”匹配，与Unicode中“0000”匹配。
把字符串设置到$W32767时一定要小心，因为执行复制时包括null。
当执行 STRCPY（W）时，字符串数据和 null 不可以超过$W32767，因为 null 被复制。（如果数据超过$W32767，将发生通信错误）

STRM2W 把字符串从 ASCII 码转换成 Unicode

适用的版本	系统版本 2 或者更高版本
格式	STRM2W (D,S)
功能	把“S”中指定的字符串从ASCII码转换成Unicode，并复制到“D”。 执行复制时包括 null。
返回值	无
例子	STRM2W（$W0，“ABC”）； ‘把“ABC”转换成 Unicode，并复制到$W0～$W2

STRW2M 把字符串从 Unicode 转换成 ASCII 码

适用的版本	系统版本 2 或者更高版本
格式	STRW2M (D,S)
功能	把“S”中指定的字符串从Unicode转换成ASCII码，并复制到“D”。 执行复制时包括 null。
返回值	无
例子	STRW2M($W0, “ABC”); “把“ABC”转换成 ASCII，并复制到$W0～$W1

SWAP　　交换指定地址的高位和低位

适用的版本	系统版本 4 或者更高版本

格式	SWAP(S,n)

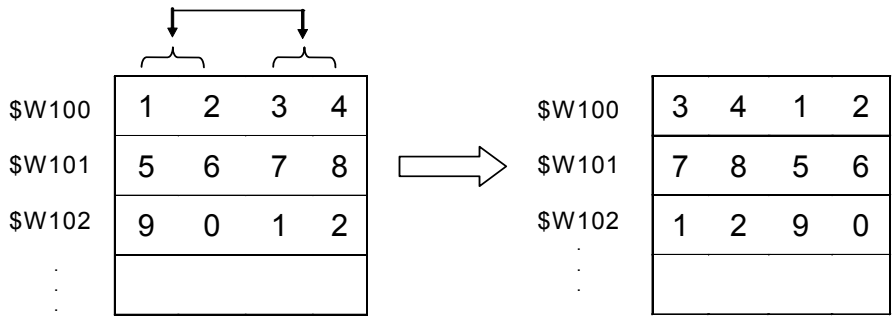
功能	交换字数据或内部保持字的高位（1字节）和低位（1字节），交换字的个数为从S开始的n个字。 S： 要交换的起始地址（$W 或 $HW）。（参阅注释） n： 要从S获取的字的数量。 设置范围如下： 当直接指定“n”时： 1 to 32767 当间接指定“n”时： $W0 to $W32767 $HW0 to $HW8191 注释：当使用为 S 指定的地址时，可以设置索引。$W 的设置范围为 0 和 32767 之间，$HW 的设置范围为 0～8191。

返回值	无

例子	SWAP（$W100,3）； 交换从$W100 获取的 3 字数据的高位和低位。

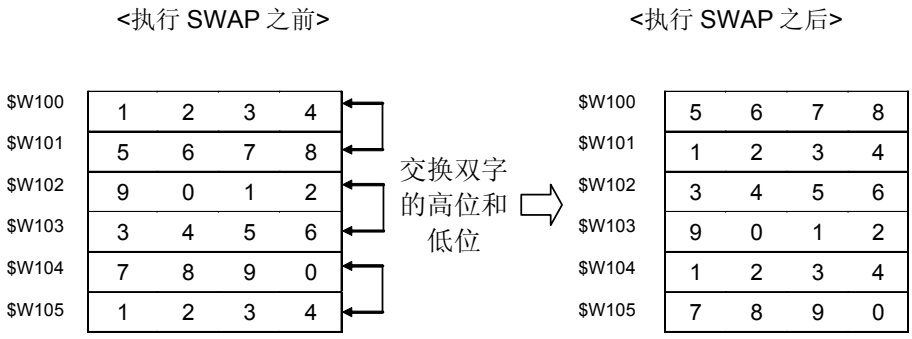
<执行 SWAP 之前>　　<执行 SWAP 之后>

交换高位和低位



SWAPL　　交换指定长型字数据的高位（2 字节）和低位（2 字节）

适用的版本	系统版本 4 或者更高版本
格式	SWAPL(S,n)
功能	交换双字数据或内部保持字（取自S的n个长型字）的高位（2字节）和低位（2字节）。 S：要交换的前列地址（$W或$HW）。（参阅注释） n：要交换的字的数量 设置范围如下： 当直接指定“n”时：1 to 32767 当间接指定“n”时：$W0 to $W32767 $HW0 to $HW8191 注释：当使用为 S 指定的地址时，可以设置索引。$W 的设置范围为 0 和 32767 之间，$HW 的设置范围为 0～8191。
返回值	无
例子	SWAPL ($W100, 3); 交换从$W100 获取的 3 个双字的高位和低位。



WRITECF 把一个 PT 存储器的内容保存到一个存储器卡（CF）中

适用的版本	系统版本 3 或者更高版本
格式	WRITECF (Mem, Size, File, Dev)
功能	把PT存储器的内容保存在存储器卡的指定文件中。指定地址的内容将以二进制格式被写入文件中。 Mem： 源数据的起始地址。可以使用（$W、$HW或$SW）索引。 $W 的设置范围为 0 和 32767 之间，$HW 的设置范围为 0 和 8191 之间。不可以指定$B、$HB 或$SB。 Size：要保存在存储器卡中的数据大小。（单位；字） 直接使用长型字、间接使用$W或$HW（使用2个字）可以设置数据的大小。设置范围 $W的设置范围为0和32767之间，$HW的设置范围为0和8191之间。如果设置值超过最大值，将发生存取错误，将出现一个错误信息。 File：目的地文件名 直接使用字符串、间接使用$W或$HW（使用2个字）可以设置数据的大小。如果为“F”指定的文件名已经存在，该文件名将被覆盖而不显示确认信息。（返回值将为0（正常完成））。可以为文件名设置多达43个数字字母混合字符（“0～9”、“A～Z”、“a～z”、“$”和“_”），包括扩展名。 Dev： 由于目的地将仅为一个存储器卡，所以总是指定 0。
返回值	0： 正常完成 －1：保存数据失败
例子	1. $W100=WRITECF($W1000, 128, “CF_FILE.BIN”, 0); 2. $W2000L=128; STRCPY($W2002, “CF_FILE.BIN”); $W100=WRITECF($W1000, $W2000, $W2002, 0);

WRITECMEM 把数据写入指定的地址

适用的版本	系统版本 2 或者更高版本
格式	WRITECMEM(D,S,n)
功能	把来自“s”的 nbit/n 通道数据复制到“D”中指定的主机中的地址。写入的最大点如下面所示。 位 126 位 字 126 通道 如果为“n”设置超过范围的值，将发生通信错误或宏执行错误。 写入的最大点取决于 PLC 类型。
返回值	无
例子	WRITECMEM([HOST1: DM0], $W0,10); ‘把$W0～$W9 复制到主机名为 HOST1 的 DM0~DM9 中。

WRITEHOSTB 把位数据写入指定的地址

适用的版本	系统版本 6.2 或者更高版本
格式	WRITEHOSTB(h, ch, addr, r, S, n)
功能	把PT存储器中S（$B/$HB）的n位数据复制到指定的主机h。 h： 主机 （主机名/主机编号） ch： 主机地址类型 *1 addr： 主机起始地址 r： 位 S： 源起始地址 （$B0 to $B32767, $HB0 to $HB8191） n： 要写入的元素的个数（1～126） * 1： 参考该章节后的*地址类型数字*
返回值	正常完成： 0x0000 出错： 高8位（B8 ～ B15）： MRES （主响应线） 低8位（B0 ～ B7） ： SRES （副响应线） * 参考用于 MRES 和 SRES 的 *5－2－7 NS－系列编程手册*中的*通信错误和措施*。
例子	把Serial A和Serial B注册到主机中： - 把$B10 ～ $B19 写入与主机 1（串行口 A）相连的 PLC 中的 CIO1000.00。 WRITEHOSTB(1, 100, 1000, 0, $B10, 10); - 把$B10 ～ $B19 写入与主机名＝［Serial B］（串行口 B）相连的 PLC 中的 DM1000.05 中。 WRITEHOSTB([SerialB], 300, 1000, 5, $HB10,10);

WRITEHOSTW 把字数据写入指定的地址

适用的版本	系统版本 6.2 或者更高版本
格式	WRITEHOSTW(h, ch, addr, r, S, n)
功能	把PT存储器中S（$B/$HB）的n个字数据复制到指定的主机h。 h： 主机 （主机名/主机编号） ch： 主机地址类型 *1 addr： 主机起始地址 r： 位 S： 源起始地址($W0 to $W32767, $HW0 to $HW8191) n： 要写入的元素的个数（1～126） * 1： 参考该章节后的*地址类型数字*
返回值	正常完成： 0x0000 出错： 高8位（B8 ～ B15）： MRES （主响应线） 低8位（B0 ～ B7） ： SRES （副响应线） * 参考用于 MRES 和 SRES 的 *5－2－7 NS－系列编程手册*中的*通信错误和措施*。
例子	把Serial A和Serial B注册到主机中： - 把$B10 ～ $B19 写入与主机 1（串行口 A）相连的 PLC 中的 CIO1000。 WRITEHOSTW(1, 100, 1000, $W10, 10); - 把$B10 ～ $B19 写入与主机名＝［Serial B］（串行口 B）相连的 PLC 中的 DM1000 中。 WRITEHOSTW([SerialB], 300, 1000, $HW10, 10);

地址类型编号

地址类型编号	地址类型名	
	BCD	二进制
PT 存储器 －$B	0	0
PT 存储器－$W	1	1
PT 存储器－$SB	2	2
PT 存储器－$SW	3	3
PT 存储器 －$HB	4	4
PT 存储器－$HW	5	5
数据区（CIO）	100	64
保持区（HR）	101	65
辅助区（AR）	102	66
链接区（LR）	103	67
工作区（WR）	104	68
定时器（TIM） 只用于 READHOSTW / WRITEHOSTW	200	C8
计数器（CNT） 只用于 READHOSTW / WRITEHOSTW	201	C9
数据存储器区（DM）	300	12C
扩展数据存储器（EM）	301	12D
扩展数据存储器 0 （EM0）	302	12E
扩展数据存储器 1 （EM1）	303	12F
扩展数据存储器 2 （EM2）	304	130
扩展数据存储器 3 （EM3）	305	131
扩展数据存储器 4 （EM4）	306	132
扩展数据存储器 5 （EM5）	307	133
扩展数据存储器 6 （EM6）	308	134
扩展数据存储器 7 （EM7）	309	135
扩展数据存储器 8 （EM8）	310	136
扩展数据存储器 9 （EM9）	311	137
扩展数据存储器 A （EMA）	312	138
扩展数据存储器 B（EMB）	313	139
扩展数据存储器 C（EMC）	314	13A

第 3 节 错误信息表

本节叙述当把宏添加到项目、屏幕和功能对象时，在错误列表框中所显示的出错信息。

3－1 错误信息表

在检查发生错误后，在错误列表框中显示的出错信息如下所示。

出错信息	说明	例子
Format error	程序包含不能识别的除变量名、功能名或编程术语以外的说明。检查输入功能是否正确。	$W0=ABC+100;
Variable name error	变量名不正确。	$B0:3=1;
(is missing	功能或句子缺少（（左括号符号）。	IF$W0==1)
No. of ( )does not agree	程序中（ ）（括号）的数量不一致。	IF($W0=1)!($W1=0
Position of , is incorrect	，（逗号）位置不正确	IF($W0==1),($W1==0)
Function argument error	程序包含不正确的功能变量，例如：把字存储器设置在只允许位存储器的位置。 参考“第2节 功能详细说明”－“功能和参数对应表”，检查参数。	$W0=BCD($B0);
= Command error	程序包含一个不正确的替换语句，例如：3＝10，$B0＝3	$W0=”ABCDE”
End of program is incomplete	输入的程序不完整	$W10=10+;

If sentence error	程序包含一个不正确的IF、ELSE或ENDIF语句	IF($W0==1)!($W1==0)
		$W10=1;
		ELSE
		$W10=10;

,or; is missing	划分语句的，（逗号）不够。程序不能被；（分号）划分。	$W10=1
FOR Statement Error	当设置FOR（n）语句时，为n设置0～32767以外的数字，或者在FOR句子中有另外一个循环结构（只能用单循环）。	FOR(50000)

		FOR(3)
		$W0=$W0+1;
		FOR(5)
		$W10=$10+10;
		NEXT;
		NEXT;