

Серия NS

Программируемые терминалы

**СПРАВОЧНОЕ
РУКОВОДСТВО ПО
МАКРОСАМ**

OMRON

Раздел 1 Функции

В данном разделе описаны условия выполнения макросов и процедура их программирования.

1-1 Что такое макросы?.....	1 – 2
1-2 Условие выполнения макроса.....	1 – 3
1-3 Программирование макросов	1 – 6

1-1 Что такое макросы?

Макрос – это функция, программируемая пользователем. Макросы расширяют возможности пользователя, предоставляя ему арифметические операции, функции сравнения и другие функции, не предусмотренные среди стандартных инструментов проектирования NS-Designer. С помощью этих функций можно управлять отображением экранов или обрабатывать данные, т.е., выполнять операции, которые ранее возлагались на ПЛК. Благодаря этому также снижается загруженность ПЛК. Событие, инициирующее выполнение макроса, называется в настоящем руководстве “условием выполнения макроса”. Классифицируя макросы по условиям выполнения, можно выделить следующие типы макросов:

- Макросы, выполняемые по событию проекта
- Макросы, выполняемые по событию экрана
- Макросы, выполняемые по событию функционального объекта

Количество макросов, которое может быть создано для одного проекта/экрана, не ограничено.

Объем макроса не должен превышать 3000 символов. Символ разрыва (перевода) строки засчитывается за два символа. Количество строк в макросе также не ограничено.

Пример:

`$W0=10;` 7 символов + перевод строки (2 символа)

`STRCPY($W10,"ABCDE");` 21 символ

В данном случае используется 30 символов.

1-2 Условие выполнения макроса

Макрос может быть создан для проекта, экрана и функционального объекта. Макрос может быть создан для одного из условий выполнения, перечисленных ниже.

Условия выполнения для проекта

Ниже перечислены условия выполнения макроса, которые могут быть использованы для проекта.

Откройте закладку [Settings (Настройка)]-[Project properties (Свойства проекта)]-[Macro (Макрос)] в NS-Designer, укажите условия выполнения и запрограммируйте макрос.

Подробное описание регистрации (назначения) макросов смотрите в *Руководстве по эксплуатации NS-Designer* (6-1 Назначение макроса - 6-1-1 Назначение макроса проекту).

Условие выполнения	Пояснение
When Loading a Project (При загрузке проекта)	Выполнение непосредственно перед загрузкой первого экрана по завершению запуска аппаратной части NS
Alarm/Event ON Timing (При возникновении тревоги/события)	Выполнение при возникновении тревоги/события
Alarm/Event OFF Timing (При отмене тревоги/события)	Выполнение при отмене тревоги/события

Условия выполнения для экрана

Ниже перечислены условия выполнения макроса, которые могут быть использованы для каждого экрана.

Откройте закладку [Settings (Настройка)]-[Screen properties (Свойства экрана)]-[Macro (Макрос)] в NS-Designer, укажите условия выполнения и запрограммируйте макрос.

Подробное описание регистрации (назначения) макросов смотрите в *Руководстве по эксплуатации NS-Designer* (6-1 Назначение макроса - Назначение макроса экрану).

Условие выполнения	Пояснение
When Loading a Screen (При открытии экрана)	Выполнение сразу после считывания данных экрана до его отображения
When Unloading a Screen (При закрытии экрана)	Выполнение непосредственно после закрытия текущего экрана

Справка

Выполнение макроса по возникновению события выполнения происходит следующим образом.

	При открытии экрана (выполнение на конечном экране)	При закрытии экрана (выполнение на конечном экране)
User screen->User screen (Экран пользователя->Экран пользователя)	Выполняется	Выполняется
User screen->Transfer screen (Экран пользователя->Экран передачи)	Не выполняется	Выполняется
User screen->System menu (Экран пользователя->Системное меню)	Не выполняется	Выполняется
System menu->User screen (Системное меню->Экран пользователя)	Выполняется	Не выполняется
User screen->Screen Saver (Экран пользователя->Сбережение экрана)	Не выполняется	Не выполняется
Screen Saver->User screen (Сбережение экрана->Экран пользователя)	Не выполняется	Не выполняется

Условия выполнения для функциональных объектов

В следующей таблице перечислены условия выполнения макроса, которые могут быть использованы для функционального объекта.

Чтобы назначить макрос функциональному объекту, откройте диалоговое окно настройки свойств для этого объекта, откройте закладку [Macro (Макрос)], выберите условие выполнения и создайте макрос.

Подробное описание регистрации (назначения) макросов смотрите в *Руководстве по эксплуатации NS-Designer* (6-1 Назначение макроса - 6-1-2 Назначение макроса функциональному объекту).

Условие выполнения	Пояснение
Touch on Timing (По нажатию)	Макрос выполняется после нажатия (прикосновения) на функциональный объект
Touch off Timing (По отжатию)	Макрос выполняется после отпускания (снятия пальца) функционального объекта, на который было произведено нажатие
Before Inputting numeral or character string (Перед вводом числового значения или символьной строки)	Макрос выполняется непосредственно перед отображением десятиклавишной или виртуальной клавиатуры для ввода значений или строк
Before Writing numeral or character string (Перед записью числового значения или символьной строки)	Макрос выполняется непосредственно перед передачей числового значения или текстовой строки в контроллер
When changing numeral and character string (При изменении числового значения или символьной строки)	Макрос выполняется после изменения числового или строкового значения по данному адресу.
When Processing Display Area (По нажатию в зоне отображения)	Макрос выполняется после нажатия на область отображения тревоги.
When Selecting an Alarm/Event (По выбору тревоги/события)	Макрос выполняется после выбора определенной строки (тревоги/события) в Табло сводки и архива тревог/событий
When selecting a list (По выбору объекта из списка)	Макрос выполняется после выбора определенной строки в списке.

Примечание

Если для функционального объекта установлен пароль, перечисленные ниже макросы выполняются после ввода пароля. Если пароль был отменен (пароль не введен), макрос выполнен не будет.

- Выполнение по нажатию/отжатию
- Выполнение перед вводом числа/строки
- Выполнение по нажатию в зоне отображения
- Выполнение при выборе тревоги/события
- Выполнение при выборе объекта из списка

Раздел 1 Введение в макросы

Серия NS - Справочное руководство по макросам

Условия выполнения, которые могут быть выбраны для функциональных объектов, перечислены в следующей таблице.

Функциональный объект	По нажатию	По отжатию	При изменении числа/строки	Перед вводом числа/строки	Перед записью числа/строки	При выборе объекта из списка
Кнопка "ВКЛ/ВЫКЛ"	OK	OK	-	-	-	-
Регистровая кнопка	OK	OK	-	-	-	-
Кнопка "Команда"	OK	OK	-	-	-	-
Одноритовая лампа	-	-	OK	-	-	-
Регистровая лампа	-	-	OK	-	-	-
Числовое поле отображения и ввода	-	-	OK	OK	OK	-
Строчное поле отображения и ввода	-	-	OK	OK	OK	-
Барабанный переключатель	-	-	OK	-	OK	-
Текст	-	-	-	-	-	-
Текстовый список	-	-	-	-	-	OK
Измеритель уровня (Вертикальная шкала)	-	-	-	-	-	-
Линейный тренд	-	-	-	-	-	-
Растровый объект	-	-	-	-	-	-
Аналоговый измеритель (Круговая шкала)	-	-	-	-	-	-
Отображение видеосигналов	-	-	-	-	-	-
Дата	-	-	-	-	-	-
Время	-	-	-	-	-	-
Тренд протокола данных	-	-	-	-	-	-
Таблица блока данных	-	-	-	OK	OK	-

[Alarm/Event object (Объект тревоги/события)]

Функциональный объект	When Pressing a Display Area (По нажатию в зоне отображения)	When selecting an Alarm/Event (По выбору тревоги/события)
Однострочное табло тревог/событий	OK	-
Табло сводки и архива тревог/событий	-	OK

1-3 Программирование макросов

В данном разделе описаны процедуры создания макросов, а также даны определения терминов, используемых при программировании.

Синтаксис макроса

Символ разделителя в программе

В конце каждой строки необходимо размещать символ деления (;). Это необязательно делать лишь для инструкций IF (), ELSEIF (), ELSE(), ENDIF.

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

Комментарий

Чтобы разместить в программе комментарий, введите перед текстом комментария одиночную кавычку ('). Весь текст после кавычки до конца строки является комментарием.

```
Пример:  
$W0 = 100; 'Комментарий  
'Комментарий  
IF ($W1==200)  
    ...
```

Термины, используемые при написании программы

При написании программы символы верхнего и нижнего регистров не различаются, поэтому можно использовать символы любого регистра.

Пример:

-MovePopwDown () и MOVEPOPWDOWN() – это одна и та же функция.
-[Host1:DM0] и [host1:dm0] – это один и тот же адрес станции.

Термины, используемые при программировании

В данном разделе описаны термины, которые используются при написании макросов.

Переменная

В тексте программы макроса могут использоваться следующие переменные.

Определение	Пояснение
Адрес в контроллере	При обращении к данным, расположенным по определенному адресу в контроллере, используйте для связи функции READCMEM и WRITECMEM. Адрес заключается в квадратные скобки [] Пример: READCMEM(\$W100,[HOST1:DM00000],100); 'Чтение из HOST1:DM00000 ... DM00099 в \$W100 ... \$W199
Память программируемого терминала	Бит Внутренняя память: \$B \$B0 ... \$B32767 (1 бит на 1 точку) Внутренняя память хранения: \$HB \$HB0 .. \$HB8191 (1 бит на 1 точку) Системная память: \$SB \$SB 47 (1 бит на 1 точку) Слово Внутренняя память: \$W \$W ... \$W32767(16 бит на 1 точку) Внутренняя память хранения: \$HW \$HW0 ... \$HW 8191 (16 бит на 1 точку) Системная память: \$SW \$SW0 ... \$SW36 (16 бит на 1 точку) Пример: \$W100=\$W0+1; 'Установка значения \$W0, добавление 1 к \$W100
Индекс (Указатель)	Для адресации битов и слов в памяти программируемого терминала (ПТ) используются указатели (индексы). Адрес с указанным в конце индексом воспринимается следующим образом: [указанный адрес + значение индекса] Всего может быть использовано 10 индексов (I0...I9). Укажите для значений \$SW27 ... \$SW36 индексы I0...I9. Пример: \$SW27 = H20; \$W0I0 = 123; '\$W0I0 воспринимается как \$W20, т.е., \$W0 + 20; \$W20 = 123

Классификатор переменной

Ниже описано применение классификаторов, которые устанавливаются для переменных. Классификаторы применяются для работы с 32-битовыми данными, а также для обращения к битам как к числу.

Параметр	Пояснение
Обращение к слову (16 бит) как к двойному слову (32 бит)	В конце переменной разместите "L". Обращение к двум словам. \$W0L=1000000; 'Обращение к \$W0, \$W1 как к 32-битовой переменной \$W100L=1000*1000; 'Обращ. к \$W100, \$W101 как к 32-битовой переменной
Обращение к битам как к числу	Разместите в конце переменной ":n". В "n" укажите значение, определяющее группу битов (до 32, с кратностью 4 бита). Исключения: Если n=16, введите "W". Если n=32, введите "L". \$B0:4 = 3; 'Запись значения 3(0011) в группу из 4 битов \$B0...\$B3 \$B0W = 12345; 'Запись 2345 (0011000000111001) в группу из 16 битов '\$B0...\$B15

Константа

В следующей таблице перечислены константы, которые могут применяться в программах макросов.

Параметр	Пояснение
Десятичная константа	При использовании слова (16 битов) можно ввести значение в диапазоне от -32768 до 32767 При использовании двух слов (32 бита) можно ввести значение от -2147483648 до 2147483647
Шестнадцатеричная константа	При использовании слова (16 бит) можно ввести значение от H0 до HFFFF При использовании двух слов (32 бита) можно ввести значение от H0 до HFFFFFFFF
Символьная строка	Заключите в кавычки (" ") Пример: "ABCDE"

Ветвления

Для определения условий можно использовать следующие ключевые слова.

Параметр	Пояснение
IF ELSEIF ELSE ENDIF	Условия выполнения, размещаемые после IF и ELSEIF, заключаются в круглые скобки (). В конце конструкции обязательно размещайте ENDIF. Допускается до 8 вложений. Количество строк в выражении IF не ограничено. Однако общее количество символов в макросе не должно превышать 3000. Пример: IF(\$W100 == 1) 'если \$W100 = 1 \$W99 = 1; ELSEIF(\$W100 == 2) 'если \$W100 = 2 \$W99 = 2; ELSE 'если \$W100 не 1 и не 2 \$W99 = 3; ENDIF

Условные выражения

Для определения условий в конструкции IF используются следующие условные выражения. Они действуют для любых типов данных (слово, слово в формате двойного слова, бит, группа битов в формате числа).

Параметр	Пояснение
A == B	Если A равно B, тогда TRUE.
A > B	Если A больше B, тогда TRUE.
A >= B	Если A больше или равно B, тогда TRUE.
A < B	Если A меньше B, тогда TRUE.
A <= B	Если A меньше или равно B, тогда TRUE.
A <> B A != B	Если A не равно B, тогда TRUE.
A && B A AND B	Если и A, и B истинны (true), тогда TRUE (AND)
A B A OR B	Если A или B истинно (true), тогда, TRUE (OR)

Справка

Результат (A&&B, A>B) может быть подставлен в качестве переменной.

Пример: \$B100=\$W0>100;

Если значение \$W = "100" или меньше, то в \$B100 будет записан "0".

Если значение \$W больше "100", в \$B100 будет записана "1".

Основные операторы

В программе можно использовать следующие операторы.

Операция	Символ	Пример	Значение
Присвоение	=	A = B	Присвоение значения B операнду A
Добавление	+	C = A+B	Запись результата A+B в C
Вычитание	-	C = A-B	Запись результата A-B в C
Умножение	*	C = A*B	Запись результата AxB в C
Деление	/	C = A/B	Запись результата A/B в C
Остаток	%	C = A%B	Запись результата A%B в C
ИЛИ		C = A B	Логическое ИЛИ операндов A и B
И	&	C = A & B	Логическое И операндов A и B
НЕ	!	C = !A	Запись в C отрицания A
Исключающее ИЛИ	^	C = A^B	Запись в C логического ИЛИ операндов A и B
Дополнение до 1	~	B = ~A	Запись в B дополнения A до 1
Сдвиг на один разряд (влево)	<<	C = A<<B	Запись в C значения, полученного после сдвига A на B битов влево
Сдвиг на один разряд (вправо)	>>	C = A>>B	Запись в C значения, полученного после сдвига A на B битов вправо

Справка

Логические операции должны выполняться над данными одного типа (например, над словами, битами или словами в формате двойных слов)

Пример: \$SW0L=\$SW10L&\$W20L; 'Обращение к словам как к двойным словам'

В одном выражении можно комбинировать несколько операций.

Пример: A=(B+C)*(D+E/2)

При выполнении операций соблюдается следующий порядок очередности.

Приоритет	Символ
Старший	()
	~
	*, /, %
	+, -
	<< , >>
	&
	^
Младший	=

Функции

Для программирования макросов в серии NS предусмотрены следующие функции.

	Действие/Операция	Функция
Преобразование форматов BCD/BIN	Значение в формате BIN ->значение в формате BCD	BCD
	Значение в формате BCD ->значение в формате BIN	BIN
Преобразование символьных строк	Копирование символьной строки	STRCPY/STRCPYW
	Преобразование из кодировки ASCII в кодировку Unicode	STRM2W
	Преобразование из кодировки Unicode в кодировку ASCII	STRW2M
Статистика тревог/событий	Обнуление количества произошедших тревог/событий	RSTALARMCNT
Специальные функции программируемого терминала	Вывод записанного значения и измененного значения	GETNUMVAL
	Переключение экрана	SHOWPAGE/SHOWPAGEBCD
	Перемещение объекта по экрану	MOVEPARTS
	Отображение диалогового окна с сообщением	MSGBOX
	Получение координат прямоугольника объекта	GETPARTS
	Перемещение всплывающего окна	MOVEPOPW
	Перемещение всплывающего окна вверх	MOVEPOPWUP
	Перемещение всплывающего окна вниз	MOVEPOPWDOWN
	Перемещение всплывающего окна влево	MOVEPOPWLEFT
	Перемещение всплывающего окна вправо	MOVEPOPWRIGHT
Обмен данными	Чтение данных по указанному адресу	READCMEM
	Запись данных по указанному адресу	WRITECMEM
	Чтение бита по указанному адресу	READHOSTB
	Чтение слова по указанному адресу	READHOSTW
	Запись бита по указанному адресу	WRITEHOSTB
	Запись слова по указанному адресу	WRITEHOSTW
Завершение процесса	Прекращение программы макроса	RETURN
Установка даты/времени	Изменение содержимого внутренних часов программируемого терминала	SETTIME

Раздел 1 Введение в макросы

Серия NS - Справочное руководство по макросам

Чтение/запись данных	Чтение содержимого (значений в двоичном формате) указанного файла в карте памяти (CF) и запись в память программируемого терминала.	READCF
	Сохранение содержимого памяти программируемого терминала в карту памяти (CF).	WRITECF
Запись одновременно по нескольким адресам	Запись (0/1) одновременно в несколько битов памяти программируемого терминала	BITSET
	Запись значения одновременно в несколько слов памяти программируемого терминала	MEMSET
Операции над данными	Перестановка местами старшего и младшего разрядов по указанному адресу.	SWAP
	Перестановка местами старших (2 байта) и младших (2 байта) разрядов указанного двойного слова.	SWAPL
	Копирование данных в пределах памяти \$W программируемого терминала	MEMCOPY
Управление "фокусом" (выделение объектов ввода)	Установка "фокуса" на указанный объект ввода	SETFOCUS
	Снятие "фокуса" с указанного объекта ввода	RELEASEFOCUS
Повторное выполнение (организация цикла)	Повторное выполнение (организация цикла)	FOR, NEXT
	Прерывание повторного (циклического) выполнения программы	BREAK
	Возвращение к началу цикла FOR.	CONTINUE

В следующей таблице перечислены события, по которым могут выполняться функции.

	Макрос проекта			Макрос экрана		Макрос функционального объекта								
	При загрузке проекта	По возникновению тревоги/события	По отмене тревоги/события	При загрузке экрана	При выгрузке экрана	По нажатию	По отжатию	Когда изменяется значение	Перед вводом числа/строки	При записи числа/строки	Текстовый список	По нажатию на область отображ.	При выборе тревоги/события	
BCD	OK	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	OK
CLOSEPOPW		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	OK
MOVEPARTS				OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
MOVEPOPW		OK	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	OK
MOVEPOPWLEFT		OK	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	OK
MOVEPOPWUP		OK	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	OK
READCMEM	OK	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	OK
RSTALARMCNT	OK	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	OK
SHOWPAGEBCD	OK	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	OK
STRM2W	OK	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	OK
SETTIME	OK	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	OK
WRITECF	OK	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	OK
SWAP	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
SWAPL	OK													
SETFOCUS		OK	OK			OK	OK	OK			OK	OK	OK	OK
RELEASEFOCUS		OK	OK			OK	OK	OK			OK	OK	OK	OK
FOR, NEXT	OK	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	OK
CONTINUE	OK	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	OK
READHOSTW	OK	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	OK
WRITE HOSTW	OK	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	OK
MEMSET	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

*Числовое поле отображения и ввода

Добавленные функции

Действие/Операция	Функция	Замечания
Установка даты/времени	SETTIME	Добавлено в NS-Designer V3.0
Чтение/запись данных	READCF	Добавлено в NS-Designer V4.0 (обозначение оборудования серии NS с окончанием – V1)
	WRITECF	
Операции над данными (перестановка)	SWAP	
	SWAPL	
	MEMCOPY	
Управление "фокусом" (выделение объектов ввода)	SETFOCUS	
	RELEASEFOCUS	
Переключение экрана (BCD)	SHOWPAGEBCD	Добавлено в NS-Designer V6.0.
Повторное выполнение (организация цикла)	FOR, NEXT	
	BREAK	
	CONTINUE	
Обмен данными	READHOSTB	Добавлено в NS-Designer V6.2.
	READHOSTW	
	WRITEHOSTB	
	WRITEHOSTW	
Запись одновременно по нескольким адресам	BITSET	
	MEMSET	

Раздел 2 Функции

В данном разделе описано применение стандартных функций.

2-1	Таблица функций и аргументов	2-2
2-2	Подробное описание функций	2-5

2-1 Таблица функций и аргументов

В следующей таблице описана номенклатура переменных и значений, которые могут быть указаны в качестве аргумента для макроса. Буквами S,D,n,x,y и т.п. в строке обозначаются параметры, которые используются в 2-2 *Подробное описание функций - Формат*.

Функция	Аргумент	Память программируемого терминала				Константа	Строка	Адрес на стороне контроллера	Указатель (индекс)
		\$B \$NB \$SB	Группа битов как число	\$W \$HW \$SW	Слово в формате двойного слова				
BCD	S		0	0	0	0		0	
BIN	S		0	0	0	0		0	
CLOSEPOPW	n		0	0	0	0		0	
GETNUMVAL	Нет								
GETPARTS	N		0	0	0	0		0	
	Left, Top, Right, Bottom			0	0			0	
MOVEPARTS	n		0	0	0	0		0	
	X		0	0	0	0		0	
	Y		0	0	0	0		0	
MOVEPOPW	n		0	0	0	0		0	
	x		0	0	0	0		0	
	y		0	0	0	0		0	
MOVEPOPWDOWN	n		0	0	0	0		0	
	y		0	0	0	0		0	
MOVEPOPWLEFT	n		0	0	0	0		0	
	x		0	0	0	0		0	
MOVEPOPWRIGHT	n		0	0	0	0		0	
	y		0	0	0	0		0	
MOVEPOPWUP	n		0	0	0	0		0	
	y		0	0	0	0		0	
MSGBOX	S1			0	0		0	0	
	S2			0	0		0	0	
	S3		0	0	0	0		0	

Функция	Аргумент	Память программируемого терминала				Константа	Строка	Адрес на стороне контроллера	Указатель (индекс)
		\$B \$NB \$SB	Группа битов как число	\$W \$HW \$SW	Слово в формате двойного слова				
READCMEM	D	0		0	0			0	
	S	0		0	0		0	0	
	N					0			
RETURN	S		0	0	0	0		0	
RSTALARMCNT	S		0	0	0	0		0	
SHOWPAGE	S		0	0	0	0		0	
SHOWPAGEBCD	S		0	0	0	0		0	
STRCPY(W)	D			0	0			0	
	S			0	0		0	0	
STRM2W	D			0	0			0	
	S			0	0		0	0	
STRM2M	D			0	0			0	
	S			0	0		0	0	
WRITECMEM	D	0		0	0		0	0	
	S	0		0	0			0	
	n	0	0	0	0				
SETTIME	S			0					
READCF	Mem			0				0	
	Size			0					
	File			0			0		
	Dev			0		0			
WRITECF	Mem			0				0	
	Size			0					
	File			0			0		
	Dev			0		0			
SWAP	S			0	0			0	
	n			0	0	0			
SWAPL	S			0	0			0	
	n			0	0	0			
MEMCOPY	S			0	0			0	
	D			0	0			0	
	N			0				0	
SETFOCUS	N			0		0		0	
RELEASEFOCUS	Нет								

READHOSTB	D	0					0
	h			0		0	0
	ch			0		0	0
	addr			0		0	0
	r			0		0	0
	n			0		0	0
READHOSTW	D			0			0
	h			0		0	0
	ch			0		0	0
	Addr			0		0	0
	n			0		0	0
WRITEHOSTB	h			0		0	0
	ch			0		0	0
	addr			0		0	0
	r			0		0	0
	S	0					0
	n			0		0	0
WRITEHOSTW	h			0		0	0
	ch			0		0	0
	addr			0		0	0
	S			0			0
	n			0		0	0
BITSET	D	0					0
	c	0				0	0
	n			0		0	0
MEMSET	D			0			0
	c			0		0	0
	n			0		0	0

2-2 Подробное описание функций

В данном разделе подробно описаны стандартные функции, которые используются для программирования макросов.

BCD Преобразование значения в формат BCD

Поддержка	Система версии 2 или выше
Формат	BCD(S)
Функция	Преобразование значения "s" в формат BCD Диапазон преобразуемых значений: от 0 до 99999999 Если указанная строка символов превышает допустимый диапазон, происходит переполнение. В конце текстовой строки размещается "0".
Возвращаемое значение	Значения в формате BCD
Пример	\$W0 = 1234; 'Запись значения 1234 в \$W0 \$W10 = BCD(\$W0); 'Запись значения в формате BCD (H1234) в \$W10 \$W20L = 12345678; 'Запись значения 12345678 в \$W20...\$W21 \$W22L = BCD(\$W20L); 'Запись значения в формате BCD (H12345678) в \$W22...W23

BIN Преобразование из формата BCD в обычный числовой формат

Поддержка	Система версии 2 или выше
Формат	BIN(S)
Функция	Преобразование значения S из формата BCD в обычный числовой формат Диапазон преобразуемых значений: от H0 до H99999999
Возвращаемое значение	Значение в числовом формате
Пример	\$W0 = H1234; 'Запись значения в формате BCD (H1234) в \$W0 \$W10 = BIN(\$W0); 'Запись 1234 в \$W10 \$W20L = H334455; 'Запись значения в формате BCD (H334455) в \$W20...\$W21 \$W22L = BIN(\$W20L); 'Запись 334455 в \$W22...\$W23

BITSET **Запись (0/1) одновременно в несколько битов в памяти программируемого терминала**

Поддержка	Система версии 6.2 или выше
Формат	BITSET(D, с, n)
Функция	<p>Запись n-го количества состояний с (0/1) в биты памяти ПТ (\$B/\$HB), начиная с бита по адресу D.</p> <p>D: Начальный адрес</p> <p>с: Записываемое значение (0/1)</p> <p>n: Количество записываемых элементов</p> <p>1...32768 (\$B)</p> <p>1...8192 (\$HB)</p>
Возвращаемое значение	Нет
Пример	<ul style="list-style-type: none"> - Запись битов 1...10, начиная с \$B100 (\$B100 ... \$B109) BITSET(\$B100, 1, 10); - Запись битов 1...10, начиная с \$B100 (\$B100 ... \$B109) \$HB100=1; \$W200=10; BITSET(\$B100, \$HB100, \$W200);

BREAK Прерывание цикла (повторяющегося выполнения программы)

Поддержка	Система версии 6 или выше
Формат	BREAK
Функция	Прерывание цикла "FOR - NEXT".
Возвращаемое значение	Нет
Пример	<p>Если \$W100I0>30=ИСТИНА, выйти из цикла FOR.</p> <pre> \$SW27=0; FOR(10) \$W100I0=\$W50I0+10; IF(\$W100I0>30) BREAK; ENDIF \$SW27=\$SW27+1; NEXT; </pre> <p>*Диапазон допустимых значений для "n": 0...32767. Отрицательное значение воспринимается как 0. В качестве адреса могут быть указаны \$W, \$HW и \$SW.</p>

CLOSEPOPW Закрытие всплывающего окна

Поддержка	Система версии 2 или выше
Формат	CLOSEPOPW(n)
Функция	<p>Закрытие всплывающего окна с указанным номером экрана "n"</p> <p>Диапазон допустимых значений для "n": 0...3999. Если всплывающий экран с указанным номером не существует, операция выполнена не будет.</p>
Возвращаемое значение	Нет
Пример	CLOSEPOPW(15); 'Закреть всплывающий экран 15

CONTINUE Повторение программы

Поддержка	Система версии 6 или выше
Формат	CONTINUE
Функция	Возвращение к началу цикла "FOR - NEXT" при выполнении данного цикла и повторное выполнение цикла "FOR - NEXT".
Возвращаемое значение	Нет
Пример	Если \$W50I0>30=ИСТИНА, произойдет возврат к началу цикла и программа цикла будет возобновлена с самого начала.

```

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

```

*Диапазон допустимых значений для "n": 0...32767. Отрицательное значение воспринимается как 0. В качестве адреса могут быть указаны \$W, \$HW и \$SW.

FOR(n), NEXT **n; количество повторов**

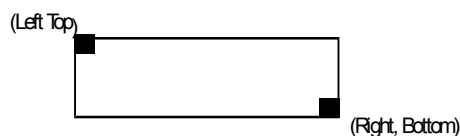
Поддержка	Система версии 6 или выше
Формат	FOR(n), NEXT n; a LOOP
Функция	Группа операторов, выполняемых циклически указанное количество раз. Цикл "FOR - NEXT" не может быть вложен в другой цикл "FOR - NEXT". (Только одинарный цикл)
Возвращаемое значение	Нет
Пример	<p>Цикл "FOR - NEXT" выполняется 10 раз, в операнды \$W0~\$W9 записывается 0 (инициализация).</p> <pre> \$W0=0; \$SW27=0; FOR(10) \$W0I0=0; \$SW27=\$SW27+1; NEXT; </pre> <p>* Диапазон допустимых значений для "n": 0...32767. Отрицательное значение воспринимается как 0. В качестве адреса могут быть указаны \$W, \$HW и \$SW.</p>

GETNUMVAL **Возврат записанного и измененного значения**

Поддержка	Система версии 2 или выше
Формат	GETNUMVAL()
Функция	Получение значения, записываемого или изменяемого с помощью числового объекта отображения и ввода. Используйте данную функцию для условий выполнения макроса "Перед записью числового значения" или "При изменении числового значения" числового объекта отображения и ввода.
Возвращаемое значение	Введенное числовое значение
Пример	<pre>\$W0=GETNUMVAL();</pre> <p>'Размещение записываемого числового значения в \$W0</p>

GETPARTS Получение координат прямоугольника объекта

Поддержка	Система версии 2 или выше
Формат	GETPARTS(n, Left, Top, Right, Bottom)
Функция	Получение координат прямоугольника для объекта с идентификатором "n". Координаты левого верхнего и правого нижнего углов прямоугольника записываются в аргументы (Left,Top) и (Right,Bottom) соответственно.

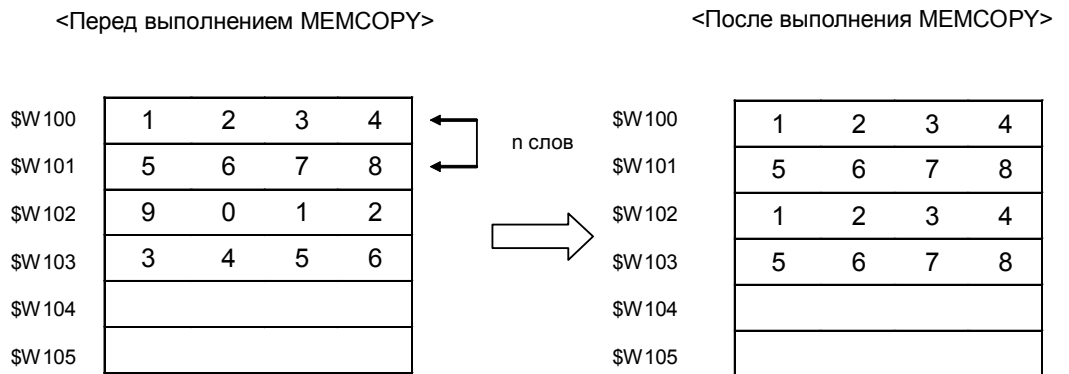


Диапазон значений "n": 0...1023. Если указано другое значение или объект с указанным идентификатором не существует, возвращается значение -1.

Возвращаемое значение	0: Выполнено без ошибок -1: Указанный объект не существует
Пример	GETPARTS(1, \$W0, \$W1, \$W2, \$W3); 'Для объекта с идентификатором 1 считываются координаты 'прямоугольника в ячейки памяти (\$W0, \$W1)-(\$W2, \$W3)

MEMCOPY Копирование данных в пределах памяти \$W программируемого терминала

Поддержка	Система версии 4 или выше
Формат	MEMCOPY (S, D, n);
Функция	<p>Копирование содержимого \$W или \$HW в памяти ПТ. S: Начальный ("верхний") адрес данных источника. D: Начальный ("верхний") адрес для записи скопированных данных. n: Количество слов данных, которое будет прочитано, начиная с S. Допускаются следующие диапазоны значений: При непосредственном указании "n": 1 ... 32767 При косвенном указании "n": \$W0 ... \$W32767 \$HW0 ... \$HW8191</p> <p>Примечание: Если для указания S используется адрес, может быть добавлен указатель (индекс). Диапазон возможных значений для \$W: 0...32767; для \$HW: 0...8191.</p>
Возвращаемое значение	Нет
Пример	<p>MEMCOPY(\$W100, \$W102, 2); Копирование двух слов из \$W 100 и запись их в \$W 102</p>



MEMSET Запись значения одновременно в несколько слов памяти программируемого терминала

Поддержка	Система версии 6.2 или выше
Формат	MEMSET(D, c, n)
Функция	<p>Запись значения c в n слов памяти ПТ (\$W/\$HW), начиная со слова с адресом D.</p> <p>D: Начальный адрес</p> <p>c: Записываемое значение</p> <p style="padding-left: 20px;">-32767...32768 (десятичный формат)</p> <p style="padding-left: 20px;">H0000...HFFFF (шестнадцатеричный формат)</p> <p>n: Количество записываемых элементов</p> <p style="padding-left: 20px;">1...32768 (\$W)</p> <p style="padding-left: 20px;">1...8192 (\$HW)</p>
Возвращаемое значение	Нет
Пример	<ul style="list-style-type: none"> - Запись значения 5 в 10 слов, начиная с \$W100 (\$W100...\$W109) MEMSET(\$W100, 5, 10); - Запись значения 5 в 10 слов, начиная с \$W100 (\$W100...\$W109) \$HW100=5; \$W200=10; MEMSET(\$W100, \$HW100, \$W200);

MOVEPARTS **Перемещение объекта в области отображения**

Поддержка	Система версии 2 или выше
Формат	MOVEPARTS (n,x,y)
Функция	<p>Перемещение объекта с идентификатором “n” в точку с указанными координатами (x, y). Указываются координаты (x, y) верхнего левого угла перемещаемого объекта. Диапазон допустимых значений для “n”: 0...1023. При вводе другого значения или при отсутствии объекта с указанным идентификатором возвращается значение –1. Значения “x, y” не ограничены. Однако значения координат x и y следует указывать таким образом, чтобы объекты оставались в пределах экрана. Значения следует указывать продуманно, поскольку при некорректных значениях некоторые или все объекты экрана могут быть удалены.</p>
Возвращаемое значение	<p>0: Выполнено без ошибок -1: Указанный объект не существует</p>
Пример	<p>MOVEPARTS (3, 150, 200); Перемещение объекта с идентификатором 3 в позицию (150,200)</p>

Справка

Если для кнопки ВКЛ/ВЫКЛ, регистровой кнопки или кнопки управления используется макрос “MOVEPARTS” и данные кнопки или данные кнопки с кадром перемещаются, необходимо выбрать условие выполнения “по отжатию”. Если выбрано “по нажатию”, тогда состояние кнопки будет определено как “нажато”.

Макрос MOVEPARTS нельзя применить для объекта отображения видео.

MOVEPOPW Перемещение всплывающего окна

Поддержка	Система версии 2 или выше
Формат	MOVEPOPW(n,x,y)
Функция	Перемещение левой верхней координаты всплывающего окна с указанным номером экрана "n" в точку с указанными координатами (x, y). Диапазон допустимых значений для "n": 0...3999. Если указано другое значение или экран с указанным номером не существует, возвращается значение "-1". Значения "x, y" не ограничены. Однако значения координат x и y следует указывать таким образом, чтобы объекты оставались в пределах экрана. При вводе некорректного значения может произойти частичное или полное удаление объектов экрана.
Возвращаемое значение	0: Выполнено без ошибок -1: Указанный экран не существует
Пример	\$W0=MOVEPOPW(10, 140, 160); 'Перемещение всплывающего экрана с номером 10 в указанную позицию (140, 160), и возврат значения "0" в \$W0. 'Если всплывающий экран не отображается, в \$W0 возвращается "-1".

MOVEPOPDOWN Перемещение всплывающего окна вниз

Поддержка	Система версии 2 или выше
Формат	MOVEPOPDOWN(n, y)
Функция	Перемещение всплывающего окна с номером экрана "n" вниз. Диапазон допустимых значений для "n": 0...3999. Если указано значение вне диапазона или экран с указанным номером не существует, возвращается значение "-1". Значение координаты "y" не ограничено. Однако значение y следует указывать таким образом, чтобы всплывающее окно оставалось в пределах экрана. При вводе некорректного значения может произойти частичное или полное удаление объектов экрана.
Возвращаемое значение	0: Выполнено без ошибок -1: Указанный экран не существует
Пример	\$W0=MOVEPOPDOWN(10, 32); 'Сдвиг всплывающего окна с номером 10 на 32 точки вниз, 'и возврат "0" в \$W0. Если всплывающий экран не отображается, 'в \$W0 возвращается "-1".

MOVEPOPWLEFT Перемещение всплывающего окна влево

Поддержка	Система версии 2 или выше
Формат	MOVEPOPWLEFT (n, x)
Функция	Перемещение всплывающего окна с номером экрана "n" влево на x точек. Диапазон допустимых значений для "n": 0...3999. Если указано значение вне диапазона или экран с указанным номером не существует, возвращается значение "-1". Значение координаты "x" не ограничено. Однако значение x следует указывать таким образом, чтобы всплывающее окно оставалось в пределах экрана. При вводе некорректного значения может произойти частичное или полное удаление объектов экрана.
Возвращаемое значение	0: Выполнено без ошибок -1: Указанный экран не существует
Пример	\$W0=MOVEPOPWLEFT (10, 32); 'Сдвиг всплывающего окна с номером 10 на 32 точки влево и возврат "0" в \$W0. Если всплывающее окно не отображается, в \$W0 возвращается значение "-1".

MOVEPOPWRIGHT Перемещение всплывающего окна вправо

Поддержка	Система версии 2 или выше
Формат	MOVEPOPWRIGHT(n, x)
Функция	Перемещение всплывающего окна с номером экрана "n" вправо на x точек. Диапазон допустимых значений для "n": 0...3999. Если указано значение вне диапазона или экран с указанным номером не существует, возвращается значение "-1". Значение координаты "x" не ограничено. Однако значение x следует указывать таким образом, чтобы всплывающее окно оставалось в пределах экрана. При вводе некорректного значения может произойти частичное или полное удаление объектов экрана.
Возвращаемое значение	0: Выполнено без ошибок -1: Указанный экран не существует
Пример	\$W0=MOVEPOPWRIGHT (10,32); 'Сдвиг всплывающего окна с номером 10 на 32 точки вправо и возврат "0" в \$W0. Если всплывающее окно не отображается, в \$W0 возвращается значение "-1".

MOVEPOPWUP Перемещение всплывающего окна вверх

Поддержка	Система версии 2 или выше
Формат	MOVEPOPWUP (n, y)
Функция	Перемещение всплывающего окна с номером экрана "n" на y точек вверх. Диапазон допустимых значений для "n": 0...3999. Если указано значение вне диапазона или экран с указанным номером не существует, возвращается значение "-1". Значение координаты "y" не ограничено. Однако значение y следует указывать таким образом, чтобы всплывающее окно оставалось в пределах экрана. При вводе некорректного значения может произойти частичное или полное удаление объектов экрана.
Возвращаемое значение	0: Выполнено без ошибок -1: Указанный экран не существует
Пример	<code>\$W0=MOVEPOPWUP (10,32);</code> 'Сдвиг всплывающего окна с номером 10 на 32 точки вверх и возврат "0" в \$W0. Если всплывающее окно не отображается, в \$W0 возвращается значение "-1".

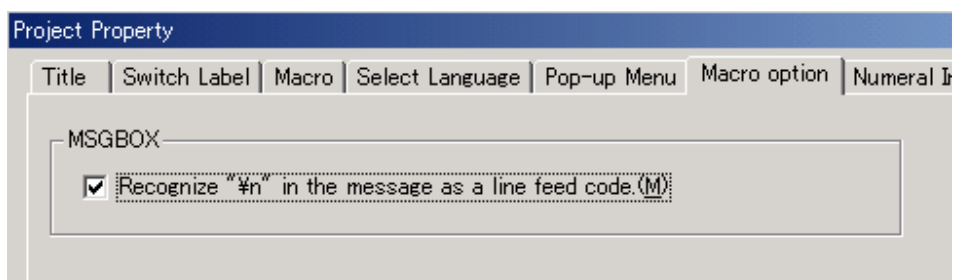
MSGBOX **Отображение диалогового окна с сообщением**

Поддержка	Система версии 2 или выше
Формат	MSGBOX (S1, S2, S3)
Функция	Отображение указанного диалогового окна с сообщением.

S1: строка сообщения

Чтобы вставить в сообщение признак разрыва строки, выполните следующие действия.

Выберите *Setting – Project Properties (Настройка – Свойства проекта)* в NS-Designer, чтобы отобразить диалоговое окно Project Property (Свойства проекта). Выберите опцию (установите флажок) “Recognize “\n” in the message as a line feed code”, если при использовании макроса отображения окна сообщения (MSGBOX) сообщения должны отображаться в виде нескольких строк, и в качестве признака разделения строки должна использоваться последовательность “\n”.



S2: текст строки заголовка

S3: ответ с пиктограммой, отображаемой в окне сообщения

Выберите тип кнопки.

8 младших битов (B0-B7)

- | | |
|--------------------------|----|
| <input type="checkbox"/> | 0: |
| ⊗ Знак СТОП | 1: |
| ⊕ Знак ВОПРОС | 2: |
| ⚠ Знак ВОСКЛИЦАНИЕ | 3: |
| ⓘ Знак ИНФОРМАЦИЯ | |

8 старших битов (B8-B15)

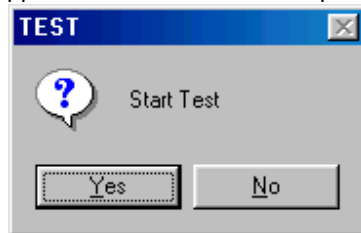
- | | |
|----|--------------------------------|
| 0: | только кнопка [OK] |
| 1: | кнопка [OK]/[Cancel] |
| 2: | кнопка [Retry]/[Cancel] |
| 3: | кнопка [Yes]/[No] |
| 4: | кнопка [Yes]/[No]/[Cancel] |
| 5: | кнопка [Stop]/[Retry]/[Ignore] |

Раздел 2 Функции

Серия NS - Справочное руководство по макросам

Возвращаемое значение	0: Выбрана кнопка [OK] 2: Выбрана кнопка [Yes] 4: Выбрана кнопка [Ignore] 6: Выбрана кнопка [Stop]	1: Выбрана кнопка [Cancel] 3: Выбрана кнопка [No] 5: Выбрана кнопка [Retry]
-----------------------	---	---

Пример	<pre> \$W0=MSGBOX ("Start Test", "TEST", H31); 'Н31: кнопка [Yes]/[No], отображение вопросительного знака IF(\$W0==2) 'Процедура записи в случае выбора "Yes" ELSE 'Процедура записи в случае выбора "No" ENDIF </pre> <p>В соответствии с произведенной настройкой отображается следующее диалоговое окно сообщения</p>
--------	--



Справка

С помощью MSGBOX может быть отображено только одно окно сообщения. При попытке отображения другого окна сообщения с помощью MSGBOX новое окно сообщения не отображается, возвращается значение "1".

Пример: создание двух однобитовых ламп

	адрес	макрос
Лампа 1	\$B0	\$W0=MSGBOX("сообщение1", "заголовок1", H31);
Лампа 2	\$B0	\$W1=MSGBOX("сообщение2", "заголовок2", H31);

Предположим, первым выполняется макрос для лампы 1. Изменение значения \$B0 приводит к отображению сообщения для лампы 1. Сообщение для лампы 2 не отображается, в \$W1 записывается значение "1".

Если в закладке *PT* системного меню включен (ON) или выключен (OFF) звуковой сигнал (Buzzer Sound) и выбрана пиктограмма СТОП или ВОСКЛИЦАНИЕ, в этом случае отображение диалогового окна сообщения сопровождается звуковым сигналом.

READCF Чтение содержимого (двоичных значений) из указанного файла карты памяти (CF) и запись в память программируемого терминала

Поддержка	Система версии 4 или выше
Формат	READCF(Mem, Size, File, Dev)
Функция	<p>Чтение содержимого (в двоичном формате) из указанного файла карты памяти в память ПТ.</p> <p>Mem: Начальный ("верхний") адрес назначения. (\$W, \$HW или \$SW). Можно указать индекс. Диапазон допустимых значений для \$W: 0...32767; для \$HW: 0...8191. \$B, \$HB и \$SB указать нельзя.</p> <p>Size: Объем читаемых данных (единицы: слово) Размер данных можно указать непосредственно в двойных словах (long) или косвенно, используя \$W или \$HW (используется 2 слова). Диапазон допустимых значений для \$W: 0...32767; для \$HW: 0...8191. Если заданное значение превышает допустимый диапазон, возникает ошибка доступа и отображается сообщение об ошибке. Если для Size указано значение 0 или меньше, в память программируемого терминала будет прочитан указанный объем данных. Если размер данных превышает значение, указанное в Size (Size>0), будет прочитан объем данных, указанный в Size. Если размер файла меньше значения, указанного в Size (Size>0), будет прочитан весь файл в том объеме, который имеется.</p> <p>File: Имя читаемого файла Имя файла может быть указано непосредственно, с помощью текстовой строки, либо косвенно с помощью \$W или \$HW (используется 2 слова). Чтение производится пословно, однако при нечетном количестве байтов в файле последний первый байт читаемого файла прочитан не будет. (Возвращается значение 0). В качестве имени файла можно ввести до 43 алфавитно-цифровых символов ("0...9", "A...Z", "a...z", "\$", "_") включая расширение.</p> <p>Dev: Адресуемое устройство. Всегда указывается 0, поскольку адресуемым устройством всегда является карта памяти.</p>
Возвращаемое значение	<p>0: Выполнено без ошибок</p> <p>-1: Сбой чтения данных</p>
Пример	<ol style="list-style-type: none"> \$W100=READCF(\$W1000,0,"CF_FILE.BIN",0); \$W2000L=0; STRCOPY(\$W2002,"CF_FILE.BIN"); \$W100=READCF(\$W1000,\$W2000,\$W2002,0);

READCMEM Чтение данных по указанному адресу

Поддержка	Система версии 2 или выше
Формат	READCMEM (D,S,n)
Функция	<p>Чтение n битов/n каналов по указанному адресу станции "S" и запись в "D". Максимальное количество точек, которое может быть прочитано:</p> <p>Бит 126 битов</p> <p>Слово 126 каналов</p> <p>Если указанное значение "n" превышает допустимый диапазон, возникает ошибка связи или ошибка выполнения макроса. Максимальное количество читаемых точек зависит от типа ПЛК.</p>
Возвращаемое значение	Нет
Пример	<p>READCMEM (\$W0, [HOST1:DM0], 10)</p> <p>'Чтение значений "DM0...DM9" из станции с именем HOST1 в ПЛК 'и запись в "\$W0...\$W9".</p>

READHOSTB Чтение битовых данных по указанному адресу

Поддержка	Система версии 6.2 или выше
Формат	READHOSTB(D, h, ch, addr, r, n)
Функция	<p>Чтение n-го кол-ва битов из станции h и запись в память ПТ (\$B/\$HB) по адресу D.</p> <p>D: Начальный адрес для записи данных (\$B0...\$B32767, \$HB0...\$HB8191)</p> <p>h: Станция (имя станции / номер станции)</p> <p>ch: Тип адреса станции *1</p> <p>addr: Начальный адрес в станции</p> <p>r: Биты</p> <p>n: Количество записываемых элементов (от 1 до 126)</p> <p>*1: См. <i>Числовые обозначения типов адреса</i> в конце данной главы.</p>
Возвращаемое значение	<p>Завершение без ошибок: 0x0000</p> <p>Ошибка: старшие 8 битов (B8...B15): MRES (главный код ответа) младшие 8 битов (B0...B7) : SRES (дополнительный код ответа)</p> <p>*Информацию о MRES и SRES смотрите в 5-2-7 <i>Ошибки связи и способы их устранения</i> в <i>Руководстве по программированию (Серия NS)</i>.</p>
Пример	<p>Станции зарегистрированы под именами Serial A и Serial B:</p> <ul style="list-style-type: none"> - Чтение десяти битов из ПЛК, подключенного к станции 1 (последовательный порт A), начиная с адреса CIO1000.00, и запись прочитанных битов в \$B10...\$B19. READHOSTB(\$B10, 1, 100, 1000, 0, 10); - Чтение десяти битов из ПЛК, подключенного к станции с именем=[Serial B] (последовательный порт B), начиная с адреса DM2000.05, и запись прочитанных битов в \$HB10...\$HB19. READHOSTB(\$HB10, [SerialB], 300, 2000, 5, 10);

READHOSTW Чтение слов данных по указанному адресу

Поддержка	Система версии 6.2 или выше
Формат	READHOSTW(D, h, ch, addr, n)
Функция	<p>Чтение n-го кол-ва слов данных из станции (h) и запись в память ПТ (\$W/\$HW) по адресу D.</p> <p>D: Начальный адрес для записи данных (\$W0...\$W32767, \$HW0...\$HW8191)</p> <p>h: Станция (имя станции / номер станции)</p> <p>ch: Тип адреса станции *1</p> <p>addr: Начальный адрес на стороне станции</p> <p>n: Количество записываемых элементов (от 1 до 126)</p> <p>*1: См. <i>Числовые обозначения типов адреса</i> в конце данной главы.</p>
Возвращаемое значение	<p>Завершение без ошибок: 0x0000</p> <p>Ошибка: старшие 8 битов (B8...B15): MRES (главный код ответа) младшие 8 битов (B0...B7) : SRES (дополнительный код ответа)</p> <p>*Информацию о MRES и SRES смотрите в 5-2-7 <i>Ошибки связи и способы их устранения</i> в <i>Руководстве по программированию (Серия NS)</i>.</p>
Пример	<p>Станции зарегистрированы под именами Serial A и Serial B:</p> <ul style="list-style-type: none"> - Чтение десяти слов данных из ПЛК, подключенного к станции 1 (последовательный порт A), начиная с CIO1000, и запись прочитанных слов в \$W10...\$W19. READHOSTW(\$W10, 1, 100, 1000, 10); - Чтение десяти слов данных из ПЛК, подключенного к станции с именем=[Serial B] (последовательный порт B), начиная с DM2000, и запись прочитанных слов в \$HW10...\$HW19. READHOSTW(\$HW10, [SerialB], 300, 2000, 10);

RELEASEFOCUS Снятие "фокуса" с объекта

Поддержка	Система версии 5 или выше
Формат	RELEASEFOCUS(); Аргументы не используются.
Функция	<p>Если на какой-либо числовой или строковый объект отображения и ввода установлен "фокус" (т.е., выделение объекта для ввода), данный макрос снимет "фокус" с объекта.</p> <ul style="list-style-type: none"> - Макрос снимет "фокус" с объекта, если объект с "фокусом" имеется на текущем отображаемом экране. - Если ни на один из объектов на текущем отображаемом экране "фокус" не установлен, данный макрос НЕ работает. - Если "фокус" установлен на объект, который располагается на самом верхнем слайде кадра, с данного объекта также будет снят "фокус". - Если "фокус" размещен на объекте накладного экрана, с данного объекта также будет снят "фокус". <p>Макрос RELEASEFOCUS НЕ срабатывает по следующим условиям выполнения.</p> <ul style="list-style-type: none"> - Во время загрузки проекта - Во время загрузки экрана - Во время считывания экрана - Перед вводом числа, набранного с помощью строковых объектов отображения и ввода. - Перед записью числового значения, введенного с помощью числового поля отображения и ввода - Перед вводом строки, набранной с помощью строковых объектов отображения и ввода - Перед записью строки, введенной с помощью строкового поля отображения и ввода
Возвращаемое значение	Нет
Пример	Во всех случаях функция снятия "фокуса" с объекта записывается следующим образом: RELEASEFOCUS();

RETURN Завершение программы макроса

Поддержка	Система версии 2 или выше
Формат	RETURN(S)
Функция	Если значение "S"="0", программа макроса прерывается и для функционального объекта продолжается дальнейшая обработка. Если введено значение, не равное "0", программа макроса завершается, дальнейшая обработка для функционального объекта прекращается.
Возвращаемое значение	Нет
Пример	RETURN(0); 'прерывание макроса и продолжение обработки RETURN(1); 'прерывание макроса и прекращение обработки

RSTALARMCNT Обнуление количества возникших тревог/событий

Поддержка	Система версии 2 или выше
Формат	RSTALARMCNT(S)
Функция	Если значение "S"=0, обнуляется количество возникших тревог. Если значение "S"=1, обнуляется количество возникших событий.
Возвращаемое значение	0: Выполнено без ошибок -1: Нет
Пример	RSTALARMCNT(0); 'обнуление количества возникших тревог RSTALARMCNT(1); 'обнуление количества возникших событий

SETFOCUS Установка "фокуса" на объект

Поддержка	Система версии 5 или выше
Формат	SETFOCUS(n);
Функция	<p>Установка "фокуса" (т.е., выделение объекта для ввода) на указанный числовой или строковый объект отображения и ввода. n: Идентификатор объекта, на который должен быть установлен "фокус". (0...32767)</p> <p>В случае установки "фокуса" на объект, указанный самым первым в списке порядка ввода (Input Order List), введите "-1".</p> <ul style="list-style-type: none"> - Данный макрос НЕ работает, если "фокус" уже установлен на другой объект. - "Фокус" НЕ будет установлен, если указанный объект находится на слайде кадра, не являющемся самым верхним. (При выполнении макроса отобразится диалоговое окно с сообщением об ошибке выполнения макроса.) - Если указанный объект не является числовым или строковым объектом отображения и ввода, данный макрос НЕ работает. - "Фокус" не может быть установлен на объект накладного экрана. <p>Макрос SETFOCUS НЕ запускается по следующим условиям выполнения.</p> <ul style="list-style-type: none"> -Во время загрузки проекта -Во время загрузки экрана - Во время считывания экрана -Перед вводом числового значения, набранного с помощью числового объекта отображения и ввода -Перед записью числового значения, введенного с помощью числового поля отображения и ввода -Перед вводом строки, набранной с помощью строкового объекта отображения и ввода -Перед записью строки, введенной с помощью строкового поля отображения и ввода
Возвращаемое значение	0: Выполнено без ошибок -1: Объект с указанным идентификатором не обнаружен.
Пример	Установка "фокуса" на объект с идентификатором 4. SETFOCUS(4);

SETTIME Изменение показаний внутренних часов программируемого терминала

Поддержка	Система версии 3 или выше
Формат	SETTIME(S)
Функция	Значения предварительно записываются по указанному адресу S, а затем записываются во внутренние часы ПТ. Диапазон допустимых значений для S: \$W0...\$W32765 или \$HW0...\$HW8189. (см. примечание) Записываемые значения должны быть представлены в формате BCD. Используется 3 слова, первое из которых находится по указанному адресу. Примечание: Для указания адреса можно использовать индекс (указатель).
Возвращаемое значение	Нет
Пример	Установка даты/времени: Декабрь 31, 2002 18:59:20. \$W 100=H5920; \$W101=H3118; \$W102=H0212; SETTIME(\$W100);

SHOWPAGE Переключение экранов

Поддержка	Система версии 2 или выше
Формат	SHOWPAGE(S)
Функция	Переключение на экран с номером "S". Диапазон допустимых значений для "S": 0...3999. Если указано значение вне диапазона, происходит ошибка выполнения макроса. Если указан номер несуществующего экрана, происходит ошибка чтения страницы (экрана).
Возвращаемое значение	Нет
Пример	SHOWPAGE(10); 'Переключение на экран с номером 10

Справка

Макрос, размещенный после SHOWPAGE, не выполняется. Обязательно размещайте SHOWPAGE в последней строке.

Пример неправильного ввода:

```
SHOWPAGE(10); <-Отобразить экран 10
$W50=100;       <-Присвоение $W50=100 не выполняется
```

Пример правильного ввода:

```
$W50=100;       <-Присвоение $W50=100 выполняется
SHOWPAGE(10); <-Отобразить экран 10
```

SHOWPAGEBCD Переключение к экрану с номером n.

Поддержка	Система версии 6 или выше
Формат	SHOWPAGEBCD(S)
Функция	S: номер экрана (H0...H3999) Переключение на экран с номером "S". Диапазон допустимых значений для "S": H0...H3999. Если значения превышают допустимый диапазон или не соответствуют формату BCD, происходит ошибка выполнения макроса. Если указан номер несуществующего экрана, происходит ошибка чтения страницы (экрана).
Возвращаемое значение	Нет
Пример	Переключение на экран 10 SHOWPAGEBCD(H10); Косвенное указание номера экрана для переключения к экрану 10. \$W100=H10; SHOWPAGEBCD(\$W100);

Справка

Макрос, размещенный после SHOWPAGEBCD, не выполняется. Обязательно размещайте SHOWPAGEBCD в последней строке.

Пример неправильного ввода:

SHOWPAGEBCD(H10); ←Переключение на страницу 10
\$W50=100; ←Присвоение \$W50=100 не выполняется

Пример правильного ввода:

\$W50=100; ←Присвоение \$W50=100 выполняется
SHOWPAGEBCD(H10); ←Переключение на страницу 10

STRCPY(W) Копирование символьной строки

Поддержка	Система версии 2 или выше
Формат	STRCPY (D,S) ;кодировка ASCII STRCPYW(D,S) ;кодировка Unicode
Функция	Копирование строки символов из D в S При копировании также копируется пустое значение (null).
Возвращаемое значение	Нет
Пример	STRCPYW (\$W0",ABC"); 'Размещение "ABC" в \$W0 to \$W2 \$W100=H6400;STRCPY(\$W110,\$W100); 'Размещение "d" в \$W110

Справка

'null' соответствует "00" в кодировке ASCII и "0000" в кодировке Unicode.
При записи строки в \$W32767 необходимо соблюдать осторожность, поскольку символ null также копируется. Копирование строковых данных и null в \$W32767 может быть не выполнено с помощью STRCPY(W), поскольку включение символа null может привести к выходу за \$W32767 и к возникновению ошибки связи.

STRM2W Преобразование символьной строки из ASCII в Unicode

Поддержка	Система версии 2 или выше
Формат	STRM2W (D,S)
Функция	Преобразование строки символов, размещенной по адресу "S", из кодировки ASCII в кодировку Unicode, и запись результата по адресу "D". При копировании также копируется пустое значение (null).
Возвращаемое значение	Нет
Пример	STRM2W(\$W0", ABC"); 'Преобразование "ABC" в Unicode и запись в \$W0...\$W2

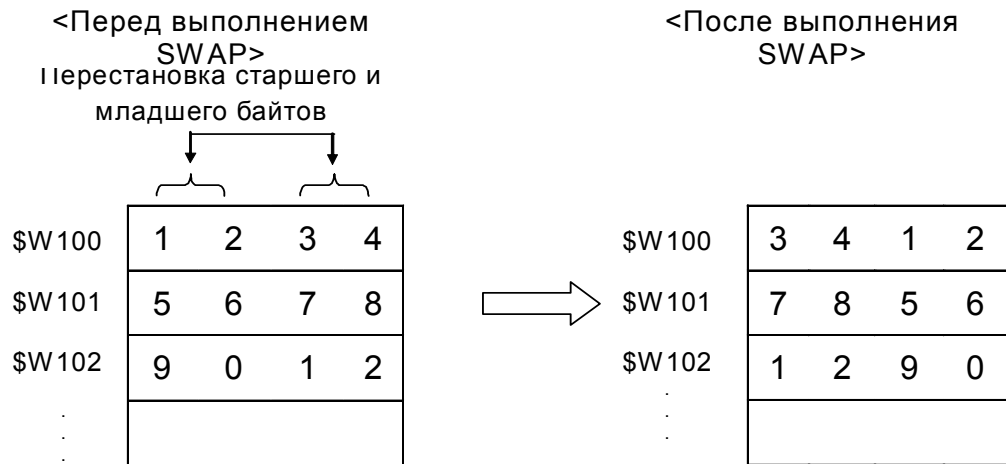
STRW2M Преобразование символьной строки из Unicode в ASCII

Поддержка	Система версии 2 или выше
Формат	STRW2M (D,S)
Функция	Преобразование строки символов, размещенной по адресу "S", из кодировки Unicode в кодировку ASCII, и запись результата по адресу "D". При копировании также копируется пустое значение (null).
Возвращаемое значение	Нет
Пример	STRW2M(\$W0, "ABC"); 'Преобразование "ABC" в ASCII и запись в \$W0...\$W1.

SWAP **Перестановка местами старших и младших разрядов по указанному адресу**

Поддержка	Система версии 4 или выше
Формат	SWAP(S,n)
Функция	<p>Перестановка местами старшего байта и младшего байта для n слов или n внутренних слов хранения, начиная со слова S. S: Старший адрес для перестановки байтов (\$W или \$HW). (см. примечание). n: Количество слов, начиная с адреса S. Допускаются следующие диапазоны значений: При непосредственном указании "n": 1 ... 32767 При косвенном указании "n": \$W0 ... \$W32767 \$HW0 ... \$HW8191</p> <p>Примечание: Если для указания S используется адрес, может быть добавлен указатель (индекс). Диапазон возможных значений для \$W: 0...32767; для \$HW: 0...8191.</p>
Возвращаемое значение	Нет

Пример	<p>SWAP(\$W100,3); Перестановка местами старшего и младшего байтов в трех словах, начиная с \$W100.</p>
--------	--



SWAPL Перестановка местами старшей и младшей пар байтов для указанного двойного слова.

Поддержка	Система версии 4 или выше
Формат	SWAPL(S,n)
Функция	<p>Перестановка старшей группы байтов (2 байта) и младшей группы байтов (2 байта) для n двойных слов или n внутренних слов хранения, начиная с S.</p> <p>S: Старший адрес для перестановки пар байтов (\$W или \$HW) (см. примечание).</p> <p>n: Количество слов для перестановки.</p> <p>Допускаются следующие диапазоны значений: При непосредственном указании "n": 1 ... 32767 При косвенном указании "n": \$W0 ... \$W32767 \$HW0 ... \$HW8191</p> <p>Примечание: Если для указания S используется адрес, можно ввести индекс (указатель). Диапазон допустимых значений для \$W: 0...32767; для \$HW: 0...8191.</p>
Возвращаемое значение	Нет
Пример	<p>SWAPL (\$W100, 3);</p> <p>Перестановка местами старшего и младшего слов (пар байтов) для трех двойных слов, начиная с \$W100.</p>



WRITECF Сохранение содержимого памяти программируемого терминала в карту памяти (CF)

Поддержка	Система версии 3 или выше
Формат	WRITECF (Mem, Size, File, Dev)
Функция	<p>Сохранение содержимого памяти ПТ в указанный файл карты памяти. Содержимое, хранящееся по указанному адресу, записывается в файл в двоичном формате.</p> <p>Mem: Начальный ("верхний") адрес данных источника (\$W, \$HW или \$SW) Может быть указан индекс.</p> <p> Диапазон допустимых значений для \$W: 0...32767; для \$HW: 0...8191. \$B, \$HB или \$SB указать нельзя.</p> <p>Size: Объем данных, сохраняемых в карту памяти (единицы: слово)</p> <p> Объем данных может быть указан непосредственно в двойных словах, либо косвенно с помощью \$W или \$HW (используется 2 слова)</p> <p> Диапазон допустимых значений для \$W: 0...32767; для \$HW: 0...8191. Если указанное значение превышает допустимый диапазон, возникает ошибка доступа и отображается сообщение об ошибке.</p> <p>File: Имя файла назначения</p> <p> Имя файла можно указать непосредственно с помощью символьной строки, либо косвенно с помощью \$W или \$HW (используется 2 слова). Если указанное имя файла "F" уже существует, файл с данным именем будет перезаписан без отображения сообщения подтверждения (возвращается значение 0 (выполнение без ошибок)). Для имени файла может быть введено до 43 алфавитно-цифровых символов ("0...9", "A...Z", "a...z", "\$", "_"), включая расширение.</p> <p>Dev: Всегда указывается 0, поскольку устройством назначения всегда является карта памяти.</p>
Возвращаемое значение	<p>0: Выполнено без ошибок</p> <p>-1: Сбой сохранения данных</p>
Пример	<ol style="list-style-type: none"> \$W100=WRITECF(\$W1000, 128, "CF_FILE.BIN", 0); \$W2000L=128; STRCPY(\$W2002, "CF_FILE.BIN"); \$W100=WRITECF(\$W1000, \$W2000, \$W2002, 0);

WRITECMEM **Запись данных по указанному адресу**

Поддержка	Система версии 2 или выше
Формат	WRITECMEM(D,S,n)
Функция	<p>Чтение содержимого n битов/n каналов из "S" и запись по указанному адресу "D" станции. Максимальное количество записываемых точек.</p> <p>Бит 126 битов</p> <p>Слово 126 каналов</p> <p>Если указанное значение "n" превышает допустимый диапазон, происходит ошибка связи или ошибка выполнения макроса. Максимальное количество записываемых точек зависит от типа ПЛК.</p>
Возвращаемое значение	Нет
Пример	<pre>WRITECMEM([HOST1: DM0], \$W0,10);</pre> <p>'Запись значений \$W0...\$W9 в станцию с именем HOST1 в ПЛК.</p>

WRITEHOSTB Запись битовых данных по указанному адресу

Поддержка	Система версии 6.2 или выше
Формат	WRITEHOSTB(h, ch, addr, r, S, n)
Функция	<p>Чтение содержимого n битов из памяти ПТ (\$B/\$HB), начиная с S, и запись в указанную станцию h.</p> <p>h: Станция (имя станции / номер станции)</p> <p>ch: Тип адреса станции *1</p> <p>addr: Начальный адрес на стороне станции</p> <p>r: Биты</p> <p>S: Начальный адрес источника (\$B0...\$B32767, \$HB0...\$HB8191)</p> <p>n: Количество записываемых элементов (от 1 до 126)</p> <p>* 1: См. <i>Числовые обозначения типов адреса</i> в конце данной главы.</p>
Возвращаемое значение	<p>Завершение без ошибок: 0x0000</p> <p>Ошибка: старшие 8 битов (B8...B15): MRES (главный код ответа)</p> <p> младшие 8 битов (B0...B7) : SRES (дополнительный код ответа)</p> <p>*Информацию о MRES и SRES смотрите в <i>5-2-7 Ошибки связи и способы их устранения</i> в <i>Руководстве по программированию (Серия NS)</i>.</p>
Пример	<p>Станции зарегистрированы под именами Serial A и Serial B:</p> <ul style="list-style-type: none"> - Запись битов \$B10...\$B19 в ПЛК, подключенный к станции 1 (последовательный порт A), начиная с CIO1000.00. WRITEHOSTB(1, 100, 1000, 0, \$B10, 10); - Запись битов хранения \$HB10...\$HB19 в ПЛК, подключенный к станции с именем=[Serial B] (последовательный порт B), начиная с DM1000.05. WRITEHOSTB([SerialB], 300, 1000, 5, \$HB10,10);

WRITEHOSTW Запись слов по указанному адресу

Поддержка	Система версии 6.2 или выше
Формат	WRITEHOSTW(h, ch, addr, r, S, n)
Функция	<p>Чтение содержимого n слов из памяти ПТ (\$W/\$HW), начиная с S, и запись в указанную станцию h.</p> <p>h: Станция (имя станции / номер станции)</p> <p>ch: Тип адреса станции *1</p> <p>addr: Начальный адрес на стороне станции</p> <p>S: Начальный адрес источника (\$W0...\$W32767, \$HW0...\$HW8191)</p> <p>n: Количество записываемых элементов (от 1 до 126)</p> <p>* 1: См. <i>Числовые обозначения типов адреса</i> в конце данной главы.</p>
Возвращаемое значение	<p>Завершение без ошибок: 0x0000</p> <p>Ошибка: старшие 8 битов (B8...B15): MRES (главный код ответа) младшие 8 битов (B0...B7) : SRES (дополнительный код ответа)</p> <p>*Информацию о MRES и SRES смотрите в 5-2-7 <i>Ошибки связи и способы их устранения</i> в <i>Руководстве по программированию (Серия NS)</i>.</p>
Пример	<p>Станции зарегистрированы под именами Serial A и Serial B:</p> <ul style="list-style-type: none"> - Запись слов \$W10...\$W19 в ПЛК, подключенный к станции 1 (последовательный порт A), начиная с CIO1000. WRITEHOSTW(1, 100, 1000, \$W10, 10); - Запись слов хранения \$HW10...\$HW19 в ПЛК, подключенный к станции с именем =[Serial B] (последовательный порт B), начиная с DM1000. WRITEHOSTW([SerialB], 300, 1000, \$HW10, 10);

Числовые обозначения типа адреса

Тип адреса	Числовое обозначение типа адреса	
	BCD	Двоичное
Память прогр. терм. - \$B	0	0
Память прогр. терм. - \$W	1	1
Память прогр. терм. - \$\$B	2	2
Память прогр. терм. - \$\$W	3	3
Память прогр. терм. - \$HB	4	4
Память прогр. терм. - \$HW	5	5
Область данных (CIO)	100	64
Область хранения (HR)	101	65
Дополнительная область (AR)	102	66
Область связей (LR)	103	67
Рабочая область (WR)	104	68
Таймер (TIM) Возможно только для READHOSTW / WRITEHOSTW	200	C8
Счетчик (CNT) Возможно только для READHOSTW / WRITEHOSTW	201	C9
Область памяти данных (DM)	300	12C
Расширенная память данных (EM)	301	12D
Расширенная память данных 0 (EM0)	302	12E
Расширенная память данных 1 (EM1)	303	12F
Расширенная память данных 2 (EM2)	304	130
Расширенная память данных 3 (EM3)	305	131
Расширенная память данных 4 (EM4)	306	132
Расширенная память данных 5 (EM5)	307	133
Расширенная память данных 6 (EM6)	308	134
Расширенная память данных 7 (EM7)	309	135
Расширенная память данных 8 (EM8)	310	136
Расширенная память данных 9 (EM9)	311	137
Расширенная память данных A (EMA)	312	138
Расширенная память данных B (EMB)	313	139
Расширенная память данных C (EMC)	314	13A

Раздел 3 Список сообщений об ошибках

В данном разделе описаны сообщения об ошибках, которые отображаются в окне списка ошибок при назначении макроса проекту, экрану и функциональному объекту.

3-1	Список сообщений об ошибках	3-2
-----	-----------------------------------	-----

3-1 Список сообщений об ошибках

В следующей таблице перечислены сообщения об ошибках, которые отображаются в окне списка ошибок при выполнении проверки на ошибки.

Сообщение об ошибке	Описание	Пример
Format error (Ошибка формата)	Программа содержит неинтерпретируемый элемент, не являющийся именем переменной, названием функции или элементом синтаксиса. Проверьте правильность введенной информации.	\$W0=ABC+100;
Variable name error (Ошибка имени переменной)	Для переменной выбрано недопустимое имя	\$B0:3=1;
(is missing ["(" отсутствует]	В функции или конструкции (выражении) отсутствует левая (открывающая) скобка "(".	IF\$W0==1)
No. of () does not agree (Кол-во открывающих и закрывающих скобок не совпадает)	Количество открывающих и закрывающих скобок в программе различается	IF(\$W0=1)!(\$W1=0
Position of , is incorrect ("," размещена некорректно)	Программа содержит некорректно установленный символ запятой ",".	IF(\$W0==1),(\$W1==0)
Function argument error (Ошибка аргумента функции)	В одной из функций программы указан некорректный аргумент. Например, для функции, оперирующей только битами, выбрана память слов. Проверьте аргумент, используя информацию в Разделе 2 "Описание функций – таблица соответствия функции и аргумента".	\$W0=BCD(\$B0);
= Command error (Ошибка команды)	Программа содержит недопустимое выражение присвоения, например, 3=10 или \$B0=3.	\$W0="ABCDE"
End of program is incomplete (Программа не завершена)	Введенная программа не завершена	\$W10=10+;
If sentence error (Ошибка выражения "If")	Программа содержит некорректное выражение (конструкцию) IF, ELSE или ENDIF	IF(\$W0==1)!(\$W1==0) \$W10=1; ELSE \$W10=10;
, or ; is missing (Отсутствует "," или ";")	Отсутствует символ запятой "," после аргумента функции. Отсутствует символ разделения строк программы ";".	\$W10=1
FOR Statement Error (Ошибка выражения FOR)	В выражении FOR(n) указано число вне диапазона 0...32767, либо в цикл FOR вложен другой цикл (вложение циклов не допускается).	FOR(50000) FOR(3) \$W0=\$W0+1; FOR(5) \$W10=\$10+10; NEXT; NEXT;

OMRON

Авторизованный дистрибьютор: