



OMRON

印刷物はありません

本 体

マクロリファレンス

プログラマブル ターミナル NSシリーズ

PT
Programmable
Terminal

形 NS15-TX01(□)-V2
形 NS12-TS0□(B)-V1/V2
形 NS10-TV0□(B)-V1/V2
形 NS8-TV□□(B)-V1/V2
形 NS5-TQ□□(B)-V2
形 NS5-SQ□□(B)-V1/V2
形 NS5-MQ□□(B)-V2
形 NSJ12-SQ0□(B)-G5D/M3D
形 NSJ10-TV0□(B)-G5D/M3D
形 NSJ8-TV0□(B)-G5D/M3D
形 NSJ5-TQ□□(B)-G5D/M3D
形 NSJ5-SQ□□(B)-G5D/M3D
形 NSH5-SQR□0B-V2

ご承諾事項

● 保証内容

(1)保証期間

本ソフトウェアの保証期間は、ご購入後またはご指定の場所に納入後1年といたします。ただし「カタログ等」に別途記載がある場合を除きます。

(2)保証範囲

- ・本ソフトウェアの使用許諾を受けたお客様が、上記保証期間中にコンピュータ・プログラムの瑕疵（マニュアルとの重要な不一致）を発見し、当社に返却した場合は、当社は瑕疵（当該マニュアルとの重要な不一致）のないコンピュータ・プログラムを記録した媒体と交換いたします。{もしくは、当社の選択により、当社ホームページより瑕疵（当該マニュアルとの重要な不一致）のないコンピュータ・プログラムをダウンロードしていただく方法により提供いたします。} また、当社の責任によるコンピュータ・プログラムの記録媒体の不良を発見し、当社に返却した場合、当社は、無償で、良品の媒体に記録したコンピュータ・プログラムと交換いたします。
- ・万一、当社がコンピュータ・プログラムの瑕疵（マニュアルとの重要な不一致）を除去できないと判断した場合は、お客様が本ソフトウェア購入代金として支払った金額をお返しいたします。

● 責任の制限

- ・前条に定める交換または購入代金の返金は、本ソフトウェアの保証責任のすべてを定めるものであり、当社は本ソフトウェアの瑕疵により発生した、お客様の直接的、間接的あるいは波及効果による損害等いかなる損害に対しても一切の責任を負いません。
- ・当社は、本ソフトウェアを当社以外の第三者が変更、改造することにより生じた瑕疵や、本ソフトウェアのインストールその他の使用によって当社商品、インストールされたソフトウェア、またはすべてのコンピュータ機器、コンピュータプログラム、ネットワーク、データベースが DDoS 攻撃（分散型 DoS 攻撃）、コンピュータウイルスその他の技術的な有害プログラム、不正アクセスを受けたことにより生じた損害につきましては、一切責任を負いません。
- ・当社は、本ソフトウェアに基づき、当社以外の第三者が開発したソフトウェアおよびそれにより生じた結果について一切の責任を負いません。

● 本ソフトウェアの用途

本ソフトウェアを本マニュアルに記載の用途以外の用途で使用しないでください。

● 仕様の変更

本ソフトウェアの仕様および付属品は改善またはその他の事由により、必要に応じて、変更される場合があります。

● 適用範囲

以上の内容は、日本国内での取引および使用を前提としております。

総目次

総目次	ii
第 1 章 マクロ機能の概要	
1-1 マクロとは	1-1
1-2 マクロ実行条件	1-2
プロジェクトの実行条件	1-2
画面の実行条件	1-3
機能部品の実行条件	1-4
1-3 マクロプログラミング	1-6
プログラム記述の方法	1-6
プログラミング用語	1-7
第 2 章 関数の説明	
2-1 関数引数対応表	2-1
2-2 関数詳細	2-3
第 3 章 エラーメッセージ一覧	
3-1 エラーメッセージ一覧	3-1

第 1 章 マクロ機能の概要

本章では、マクロ機能を使用する上で必要な実行条件やプログラミング方法などの共通的な内容について説明します。

1-1 マクロとは	1 - 1
1-2 マクロ実行条件	1 - 2
1-3 マクロプログラミング	1 - 6

1-1 マクロとは

マクロ機能とは、ユーザ独自の簡単なプログラムを作成して実行できる機能です。算術演算や、条件の判別など、CX-Designer の基本機能でサポートされていない機能を自由に追加することができます。

PLC で行っていた画面の表示処理やデータ処理などを、マクロ機能を使用して NS 本体側での処理に移行することができます。これにより PLC 側の負荷を軽減することができます。

本書において、マクロを実行するタイミングのことを、マクロの実行条件と呼びます。

マクロは、大きく分けて以下の 3 つの実行条件にて作成することができます。

- ・プロジェクトの実行条件
- ・画面の実行条件
- ・機能部品の実行条件

1 プロジェクト／1 画面に作成できるマクロの数に制約はありません。

1 マクロで入力可能な文字数は全半角あわせて 3000 文字です。改行は 2 文字として数えます。行数としての制約はありません。

例

‘マクロ入力可能文字数 …11 文字+改行(2 文字)、コメント文もカウント。

\$W0=10; …7 文字+改行(2 文字)

STRCPY(\$W10,”あいうえお”); …21 文字

このマクロの場合、43 文字使用したことになります。

1-2 マクロ実行条件

マクロ機能はプロジェクト、画面、および機能部品のそれぞれに対して作成することができ、また以下に示すそれぞれの実行条件ごとに作成することができます。

プロジェクトの実行条件

プロジェクトに対して作成可能なマクロの実行条件は以下のとおりです。

CX-Designer では、[PT]－[プロジェクトプロパティ設定]－[マクロ]タブ画面にて実行条件の設定およびマクロの作成を行います。

操作の詳細については、CX-Designer オンラインヘルプ「システムとプロジェクトの設定を行う」をご参照ください。

マクロ実行条件	説 明
プロジェクトロード時	NS 本体起動後、最初の画面をロードする直前に実行されます。
アラーム／イベント ON 時	アラームが発生した時に実行されます。
アラーム／イベント OFF 時	アラームが解除された時に実行されます。
ビット変化時	指定したビットタイプの通信アドレスの変化時に実行するマクロを設定します。最大 10 個まで設定できます。
値変化時	指定したワードタイプの通信アドレスの変化時に実行するマクロを設定します。最大 10 個まで設定できます。

画面の実行条件

各画面に対して作成可能なマクロの実行条件は以下のとおりです。

CX-Designer では、[PT]－[画面/シートプロパティ]－[マクロ]タブ画面にて実行条件の設定およびマクロの作成を行います。

操作方法の詳細については、CX-Designer オンラインヘルプ「画面を作成する」をご参照ください。

マクロ実行条件	説 明
画面ロード時	次に表示する画面データを読みこんだ直後に実行されます。
画面アンロード時	現在の画面を閉じた直後に実行されます。

MEMO

◆ 画面マクロは以下のタイミングで実行されます。

	画面ロード時 (移行先画面での実行)	画面アンロード時 (移行元画面での実行)
ユーザ画面→ユーザ画面	実行される	実行される
ユーザ画面→転送画面	実行されない	実行される
ユーザ画面→システムメニュー	実行されない	実行される
システムメニュー→ユーザ画面	実行される	実行されない
ユーザ画面→スクリーンセーバ	実行されない	実行されない
スクリーンセーバ→ユーザ画面	実行されない	実行されない

機能部品の実行条件

各機能部品に対して作成可能なマクロの実行条件は以下のとおりです。

CX-Designer では、各機能部品の[プロパティダイアログ]－[マクロ]タブ画面にて実行条件の設定およびマクロの作成を行います。

操作方法の詳細については、CX-Designer オンラインヘルプ「機能部品を作成する」をご参照ください。

マクロ実行条件	説 明
タッチ ON 時	機能部品を押した瞬間に実行されます。
タッチ OFF 時	機能部品を押して離れた瞬間に実行されます。
数値・文字列入力開始前	数値や文字列を入力するためのテンキーや仮想キーボードが表示される直前に実行されます。
数値・文字列書込み前	入力した数値、文字列をホストへ通知する直前に実行されます。
値変化時 (数値・文字列変化・数値比較)	通信アドレスの値が変化したときに実行されます。
表示エリア押下時	アラーム表示の表示エリアを押した瞬間に実行します。
アラーム／イベント選択時	アラーム／イベントサマリ・履歴に表示された各アラーム／イベントを選択した直後に実行されます。
リスト選択時	リスト選択に表示されたリストを選択した直後に実行されます。



機能部品にパスワードが設定されている場合、以下のマクロは、パスワード入力後に実行され、また、パスワード入力キャンセル時は実行されません。

- ・ タッチ ON 時
- ・ タッチ OFF 時
- ・ 数値・文字列入力開始前
- ・ 表示エリア押下時
- ・ アラーム／イベント選択時
- ・ リスト選択時

各機能部品で選択できる実行条件は以下のとおりです。

機能部品	タッチ ON 時	タッチ OFF 時	(数値・文字列変化時・ 数値比較)	数値・文字列 入力開始前	数値・文字列 書き込み前	リスト選択時
ON/OFF ボタン	○	○	—	—	—	—
ワードボタン	○	○	—	—	—	—
コマンドボタン	○	○	—	—	—	—
ビットランプ	—	—	○	—	—	—
ワードランプ	—	—	○	—	—	—
数値表示&入力	—	—	○	○	○	—
文字列表示&入力	—	—	○	○	○	—
サムロータリスイッチ	—	—	○	—	○	—
ラベル	—	—	—	—	—	—
リスト選択	—	—	—	—	—	○
レベル表示	—	—	—	—	—	—
折れ線グラフ	—	—	—	—	—	—
ビットマップ	—	—	—	—	—	—
アナログメータ	—	—	—	—	—	—
ビデオ表示	—	—	—	—	—	—
日付	—	—	—	—	—	—
時刻	—	—	—	—	—	—
データロググラフ	—	—	—	—	—	—
データブロックテーブル	—	—	—	○	○	—
テンポラリー入力	—	—	—	—	—	—
連続線描画	—	—	—	—	—	—
マルチファンクション	○	○	○	—	—	—
コンテンツ表示	—	—	—	—	—	—

[アラーム／イベント部品]

機能部品	表示エリア押下時	アラーム／イベント選択時
アラーム／イベント表示	○	—
アラーム／イベント履歴・サマリ	—	○

MEMO

◆ コマンドボタンの機能を以下に設定した場合、タッチ ON 時に設定されたマクロはタッチ OFF 時に実行されます。OFF 時マクロに設定されたマクロは実行されません。

- ・画面切換
- ・ポップアップ画面制御
- ・システムメニュー表示
- ・データブロック制御

上記以外の機能を選択した場合は、指定したタイミングでマクロが実行されます。

1-3 マクロプログラミング

マクロプログラムの記述方法と、使用できるプログラミング用語について説明します。

プログラム記述の方法

プログラムの区切り

各プログラムの区切りとして最後にセミコロン(;)を記述します。ただし、IF()、ELSEIF()、ELSE、ENDIFにはセミコロンを付けません。

例:

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

コメント文

各プログラムにコメントを記述する場合は、クォーテーション(‘)記号を文頭に記述します。クォーテーションから行の末尾までがコメントとみなされます。

例:

```
$W0 = 100; ‘ コメント文  
‘コメント文  
IF($W1== 200)  
    . . . .
```

プログラミング用語の記述

プログラムで使用するプログラミング用語は大文字と小文字が区別されないので、どちらで記述しても問題ありません。

また、分かりやすいように大文字と小文字を混在して記述しても、正常に処理されます。

ただし、タグ名に全角文字を使用している場合は、大文字、小文字は区別されます。また全角文字、半角文字も区別されます。

例

- MovePopwDown()と MOVEPOPWDOWN()は同じ関数として処理されます。
- [Host1:DM0]と[host1:dm0]は同じホストの通信アドレスとして処理されます。
- [HOST1:TAG1](半角)、[HOST1:TAG1](全角)、[HOST1:tag1](全角)はすべて別のタグとして処理されます。

プログラミング用語

本機能で使用する用語について説明します。

変数

マクロプログラムで使用可能な変数は以下のとおりです。

項 目	説 明
ホストのアドレス	ホストの通信アドレス、タグは [] で囲って記述します。 ホストとの通信用の関数 READCMEM,WRITECMEM で使用できます。 例: <code>READCMEM (\$ W100,[HOST1:DM00000],100);</code> <code>'HOST1:DM00000~HOST1:DM00099 を \$ W100~ \$ W199 に読み出し</code>
PT メモリ	ビット変数 内部メモリ: \$ B \$ B0~ \$ B32767 (1 点当たり 1 ビット) 内部保持メモリ: \$ HB \$ HB0~ \$ HB8191(1 点当たり 1 ビット) システムメモリ: \$ SB \$ SB0~ \$ SB47 (1 点当たり 1 ビット) ワード変数 内部メモリ: \$ W \$ W 0~ \$ W32767 (1 点当たり 16 ビット) 内部保持メモリ: \$ HW \$ HW0~ \$ HW8191(1 点当たり 16 ビット) システムメモリ: \$ SW \$ SW 0~ \$ SW36 (1 点当たり 16 ビット) 例: <code>\$ W100 = \$ W0+1;</code> ‘ \$ W0 の値に 1 を加えた値を \$ W100 にセット
インデックス変数	PT メモリのビット変数、ワード変数に対してインデックス修飾するための変数です。通信アドレスの末尾に付加することにより、[指定通信アドレス+インデックス変数内の値]の通信アドレスに対する処理を行います。 10~19 の 10 点があります。 例: <code>\$ SW27 = H20;</code> <code>\$ W0I0 = 123;</code> ‘ \$ W0I0 は \$ W0 のアドレスに 20 加えた \$ W20 として扱われる ‘ \$ W20 = 123

変数の修飾子

変数の修飾子は以下のように記述します。

修飾子は、32 ビットデータに対する処理を行う場合やビット変数に対して数値処理を行う場合に使用します。

項 目	説 明
ワード変数(16 ビット)のロング(32 ビット)アクセス	変数名の後に L を付けます。2 ワードを使用します。 \$ W0L = 1000000; ‘ \$ W0, \$ W1 を 32 ビットとしてアクセス \$ W100L = 1000*1000; ‘ \$ W100, \$ W101 を 32 ビットとしてアクセス
ビット変数の数値アクセス	変数名の後に :n を付けます。n にはビット数を指定し、4 ビット単位で最大 32 まで指定可能です。特殊な記述方式として n=16 のときは W、n=32 のときは L を付けます。 \$ B0:4 = 3; ‘ \$ B0 ~ \$ B3 の 4 ビットへ 3(0011)をセット \$ B0W = 12345; ‘ \$ B0 ~ \$ B15 の 16 ビットへ 12345(0011000000111001)をセット

定数

マクロプログラムで使用可能な定数と、それぞれの記述方法は以下のとおりです。

項 目	説 明
10 進定数	ワードデータ(16 ビット)使用時-32768~32767 2 ワードデータ(32 ビット)使用時-2147483648~2147483647
16 進定数	ワードデータ(16 ビット)使用時 H0~HFFFF 2 ワードデータ(32 ビット)使用時 H0~HFFFFFFFF
文字列	“”で括ります。 例: “ABCDE”

条件分岐

条件判別は以下のように記述します。

項 目	説 明
IF ELSEIF ELSE ENDIF	IF、ELSEIF に続けてカッコ () で条件式を囲みます。必ず最後は ENDIF を記述します。入れ子は、8 つまで記述することができます。 IF 文内に入力可能な行数に制限はありません。ただし、マクロ全体の入力文字数が 3000 文字以内である必要があります。 例: IF (\$W100 == 1) ‘ \$W100 が 1 なら \$W99 = 1; ELSEIF (\$W100 == 2) ‘ \$W100 が 2 なら \$W99 = 2; ELSE ‘ \$W100 が 1 と 2 以外なら \$W99 = 3; ENDIF

条件式

IF 文で条件判別する際の条件式は、以下のように記述します。PT メモリ (\$B/\$HB/\$SB/\$W/\$HW/\$SW) のすべてのデータ型(ワード変数、ワード変数のロングアクセス、ビット変数、ビットの数値アクセス)で使用できます。

項 目	説 明
A == B	A と B が等しいとき成立
A > B	A が B より大きいとき成立
A >= B	A が B 以上のとき成立
A < B	A が B より小さいとき成立
A <= B	A が B 以下のとき成立
A <> B A != B	A と B が異なるとき成立
A && B A AND B	A の条件が成立かつBの条件が成立のとき成立 (論理積)
A B A OR B	A の条件が成立またはBの条件が成立のとき成立 (論理和)

MEMO

- ◆ 条件の判定 (A&&B、A>B など) の結果を変数に代入することもできます。

例: \$B100=\$W0 > 100;

\$W の値が“100”以下のとき \$B100 には“0”が、\$W の値が“101”以上のとき \$B100 には“1”が代入されます。

- ◆ インデックス指定も可能です。

例:

\$SW27 = H100;

‘\$W0i0 は \$W0 のアドレスに 100 加えた \$W100 として扱われる

\$W0i0 = 120;

‘\$W100 = 120

IF(\$W0i0 > 100)

‘\$W100 の値が“101”以上なら

\$B0i0 = 1;

ELSE

‘\$W100 の値が“100”以下なら

\$B0i0 = 0;

ENDIF

- ◆ ホスト側アドレスを使用することはできません。

基本演算命令

プログラム内で使用できる基本演算命令は以下のとおりです。

項 目	シンボル	使用例	意 味
代入	=	A = B	A に B を代入
加算	+	C = A + B	A+B を C にセット
減算	-	C = A - B	A-B を C にセット
乗算	*	C = A * B	A×B を C にセット
除算	/	C = A / B	A÷B を C にセット
剰余	%	C = A % B	A と B の剰余を C にセット
論理和 (OR)		C = A B	A と B の論理和を C にセット
論理積 (AND)	&	C = A & B	A と B の論理積を C にセット
否定 (NOT)	!	C = ! A	A の否定を C にセット
排他論理和	^	C = A ^ B	A と B の排他論理和を C にセット
1の補数	~	B = ~ A	A の 1 の補数 (ビット反転) を B にセット
ビットシフト(左)	<<	C = A << B	A を B ビット分左へ算術シフト処理した値を C へセット
ビットシフト(右)	>>	C = A >> B	A を B ビット分右へ算術シフト処理した値を C へセット

MEMO

- ◆ 論理演算を実行する場合は、同じデータ型同士(ワード変数同士、ビット変数同士、ロングアクセス同士)で行ってください。

例: \$W0L=\$W10L&\$W20L; ' 全てワード変数のロングアクセスを使用

また、複数の演算を組み合わせることも可能です。

例:

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

演算の優先度は以下のとおりです。

項 目	シンボル
高	()
↑	~, !
	*, / , %
	+, -
	<< , >>
	&
	^
↓	
低	=

関数一覧

NS シリーズのマクロ機能で準備されている関数は以下のとおりです。

	機 能	関数名
BCD $\leftrightarrow$ BIN 変換	数値 (BIN コード) $\rightarrow$ BCD コード	BCD
	BCD コード $\rightarrow$ 数値 (BIN コード)	BIN
文字列操作	文字列コピー	STRCPY/STRCPYW
	マルチバイト文字列をユニコード文字列に変換	STRM2W
	ユニコード文字列をマルチバイト文字列に変換	STRW2M
アラーム/イベント履歴	アラーム/イベント発生回数を 0 クリア	RSTALARMCNT
HMI 専用命令	数値書込み値、変化値の取り出し	GETNUMVAL
	画面切換え	SHOWPAGE/ SHOWPAGEBCD
	部品表示エリア移動	MOVEPARTS
	メッセージダイアログボックス表示	MSGBOX
	部品の表示矩形取得	GETPARTS
	ポップアップウィンドウ画面移動	MOVEPOPW
	ポップアップウィンドウ画面を上へ移動	MOVEPOPWUP
	ポップアップウィンドウ画面を下へ移動	MOVEPOPWDOWN
	ポップアップウィンドウ画面を左へ移動	MOVEPOPWLEFT
	ポップアップウィンドウ画面を右へ移動	MOVEPOPWRIGHT
	ポップアップウィンドウ画面を閉じる	CLOSEPOPW
通信	指定通信アドレスからのデータ読出し	READCMEM
	指定通信アドレスへのデータ書込み	WRITECMEM
	指定通信アドレスからのビットデータ読出し	READHOSTB
	指定通信アドレスからのワードデータ読出し	READHOSTW
	指定通信アドレスへのビットデータ書込み	WRITEHOSTB
	指定通信アドレスへのワードデータ書込み	WRITEHOSTW
終了処理	マクロプログラムの終了	RETURN
時計設定	NS 本体の内部時計を変更	SETTIME
データ読み書き	メモ리카ード (CF) 上の指定ファイルの内容 (バイナリ値) を PT メモリに読み出し	READCF
	PT メモリの内容をメモ리카ード (CF) 上の指定ファイルに保存	WRITECF
データ一括書込み	PT メモリのビットアドレスに (0/1) を一括で書込む	BITSET
	PT メモリのワードアドレスに値を一括で書込む	MEMSET
データ操作・変換	指定アドレスの上位バイト、下位バイトを入れ替え	SWAP
	指定アドレスの上位バイト、下位バイトをロングワード単位で入れ替え	SWAPL
	PT メモリ \$ W の内容をコピー	MEMCOPY
	ローカル時間 (ns 単位) をカレンダー時間 (年月日時分秒) に変換	LOCALTIME
フォーカス制御	指定部品 ID 番号の部品に入力フォーカスをセットする	SETFOCUS
	セットされているフォーカスを解除する	RELEASEFOCUS
プログラムの繰り返し	プログラムの繰り返し	FOR,NEXT
	プログラムの繰り返しを中断する	BREAK
	プログラムの繰り返し対象の先頭に戻る	CONTINUE

数学演算マクロ機能	逆余弦(アークコサイン)関数	ACOS
	逆正弦(アークサイン)関数	ASIN
	逆正接(アークタンジェント)関数	ATAN
	2変数のアークタンジェント関数	ATAN2
	切り上げ関数	CEIL
	余弦(コサイン)関数	COS
	度単位の数値をラジアン単位に変換する関数	DEG2RAD
	e の累乗を計算する	EXP
	切捨て関数	FLOOR
	自然対数関数	LOG
	値がゼロに近い値にでも精度を保つ方法で計算した $\log(1+\text{number})$ を返す関数	LOG10
	ラジアン単位の数値を度単位に変換する関数	RAD2DEG
	平方根関数	SQRT
	正弦(サイン)関数	SIN
	正接(タンジェント)関数	TAN
FLOAT 演算マクロ機能	指数関数値	POW
	2つの FLOAT 精度の数値を加算する関数	FADD
	2つの FLOAT 精度の数値を比較する関数	FCOMP
	2つの FLOAT 精度の数値を除算する関数	FDIV
	2つの FLOAT 精度の数値を乗算する関数	FMUL
	2つの FLOAT 精度の数値を減算する関数	FSUB
	FLOAT 精度の数値をセットする関数	FSET

また、各関数を実行できるタイミングは以下のとおりです。

	プロジェクト マクロ			画面 マクロ		機能部品マクロ							
	プロジェクト ロード時	アラーム/ イベントON時	アラーム/ イベントOFF時	画面 ロード時	画面 アンロード時	タッチ ON時	タッチ OFF時	値 変化時	数値・文字 列入力開始前	数値・文字 列書込み前	リスト 選択時	表示エリア 押下時	アラーム/ イベント選択時
BCD	○	○	○	○	○	○	○	○	○	○	○	○	○
BIN	○	○	○	○	○	○	○	○	○	○	○	○	○
CLOSEPOPW	×	○	○	○	○	○	○	○	○	○	○	○	○
GETNUMVAL	×	×	×	×	×	×	×	△	△	△	×	×	×
GETPARTS	×	×	×	○	○	○	○	○	○	○	○	○	○
MOVEPARTS	×	×	×	○	○	○	○	○	○	○	○	○	○
MOVEPOPW	×	○	○	○	○	○	○	○	○	○	○	○	○
MOVEPOPWDOWN	×	○	○	○	○	○	○	○	○	○	○	○	○
MOVEPOPWLEFT	×	○	○	○	○	○	○	○	○	○	○	○	○
MOVEPOPWRIGHT	×	○	○	○	○	○	○	○	○	○	○	○	○
MOVEPOPWUP	×	○	○	○	○	○	○	○	○	○	○	○	○
MSGBOX	○	○	○	○	○	○	○	○	○	○	○	○	○
READCMEM	○	○	○	○	○	○	○	○	○	○	○	○	○
RETURN	○	○	○	○	○	○	○	○	○	○	○	○	○
RSTALARMCNT	○	○	○	○	○	○	○	○	○	○	○	○	○
SHOWPAGE	○	○	○	×	×	○	○	○	○	○	○	○	○
SHOWPAGEBCD	○	○	○	×	×	○	○	○	○	○	○	○	○
STRCPY(W)	○	○	○	○	○	○	○	○	○	○	○	○	○
STRM2W	○	○	○	○	○	○	○	○	○	○	○	○	○
WRITECMEM	○	○	○	○	○	○	○	○	○	○	○	○	○
SETTIME	×	○	○	○	○	○	○	○	○	○	○	○	○
READCF	○	○	○	○	○	○	○	○	○	○	○	○	○
WRITECF	○	○	○	○	○	○	○	○	○	○	○	○	○
MEMCOPY	○	○	○	○	○	○	○	○	○	○	○	○	○
SWAP	○	○	○	○	○	○	○	○	○	○	○	○	○
SWAPL	○	○	○	○	○	○	○	○	○	○	○	○	○
SETFOCUS	×	○	○	×	×	○	○	○	×	×	○	○	○
RELEASEFOCUS	×	○	○	×	×	○	○	○	×	×	○	○	○
FOR,NEXT	○	○	○	○	○	○	○	○	○	○	○	○	○
BREAK	○	○	○	○	○	○	○	○	○	○	○	○	○
CONTINUE	○	○	○	○	○	○	○	○	○	○	○	○	○
READHOSTB	○	○	○	○	○	○	○	○	○	○	○	○	○
READHOSTW	○	○	○	○	○	○	○	○	○	○	○	○	○
WRITEHOSTB	○	○	○	○	○	○	○	○	○	○	○	○	○
WRITEHOSTW	○	○	○	○	○	○	○	○	○	○	○	○	○
BITSET	○	○	○	○	○	○	○	○	○	○	○	○	○
MEMSET	○	○	○	○	○	○	○	○	○	○	○	○	○
ACOS	○	○	○	○	○	○	○	○	○	○	○	○	○
ASIN	○	○	○	○	○	○	○	○	○	○	○	○	○
ATAN	○	○	○	○	○	○	○	○	○	○	○	○	○
ATAN2	○	○	○	○	○	○	○	○	○	○	○	○	○
CEIL	○	○	○	○	○	○	○	○	○	○	○	○	○

NS シリーズ マクロリファレンス

COS	○	○	○	○	○	○	○	○	○	○	○	○	○
DEG2RAD	○	○	○	○	○	○	○	○	○	○	○	○	○
EXP	○	○	○	○	○	○	○	○	○	○	○	○	○
FLOOR	○	○	○	○	○	○	○	○	○	○	○	○	○
LOG	○	○	○	○	○	○	○	○	○	○	○	○	○
LOG10	○	○	○	○	○	○	○	○	○	○	○	○	○
RAD2DEG	○	○	○	○	○	○	○	○	○	○	○	○	○
SQRT	○	○	○	○	○	○	○	○	○	○	○	○	○
SIN	○	○	○	○	○	○	○	○	○	○	○	○	○
TAN	○	○	○	○	○	○	○	○	○	○	○	○	○
POW	○	○	○	○	○	○	○	○	○	○	○	○	○
FADD	○	○	○	○	○	○	○	○	○	○	○	○	○
FCOMP	○	○	○	○	○	○	○	○	○	○	○	○	○
FDIV	○	○	○	○	○	○	○	○	○	○	○	○	○
FMUL	○	○	○	○	○	○	○	○	○	○	○	○	○
FSUB	○	○	○	○	○	○	○	○	○	○	○	○	○
FSET	○	○	○	○	○	○	○	○	○	○	○	○	○
LOCALTIME	○	○	○	○	○	○	○	○	○	○	○	○	○

△: 数値表示&入力のみで可能

追加関数一覧

機能	関数名	備考
時計設定	SETTIME	NS 本体システムバージョン Ver.3.0 より追加
データ読み書き	READCF	NS 本体システムバージョン Ver.4.0 より追加
	WRITECF	
データ操作・変換	SWAP	
	SWAPL	
	MEMCOPY	
フォーカス制御	SETFOCUS	NS 本体システムバージョン Ver.5.0 より追加
	RELEASEFOCUS	
画面切替(BCD)	SHOWPAGEBCD	NS 本体システムバージョン Ver.6.0 より追加
プログラムの繰り返し	FOR,NEXT	
	BREAK	
	CONTINUE	
通信	READHOSTB	NS 本体システムバージョン Ver.6.2 より追加
	READHOSTW	
	WRITEHOSTB	
	WRITEHOSTW	
データ一括書込み	BITSET	NS 本体システムバージョン Ver.8.2 より追加
	MEMSET	
算術演算	ACOS	
	ASIN	
	ATAN	
	ATAN2	
	CEIL	
	COS	
	DEG2RAD	
	EXP	
	FLOOR	
	LOG	
	LOG10	
	RAD2DEG	
	SQRT	
	SIN	
	TAN	
	POW	
FLOAT 演算マクロ機能	FADD	
	FCOMP	
	FDIV	
	FMUL	
	FSUB	
	FSET	
データ操作・変換	LOCALTIME	NS 本体システムバージョン Ver.8.5 より追加

第2章 関数の説明

本章では基本関数の使用方法について詳しく説明します。

2-1 関数引数対応表	2 - 1
2-2 関数詳細	2 - 3

2-1 関数引数対応表

マクロ関数で引数として指定できる変数や値の種類は以下のとおりです。

なお、引数列の S,D,n,x,y などのアルファベットは「2-2 関数詳細」の「書式」で使用しているパラメータをあらわしています。

関数名	引数	PT メモリ				定数	文字列	ホスト側 アドレス	INDEX 指定
		\$B \$HB \$SB	ビット変数 の数値 アクセス	\$W \$HW \$SW	ワード変数 のロング アクセス				
BCD	S		○	○	○	○			○
BIN	S		○	○	○	○			○
CLOSEPOPW	n		○	○	○	○			○
GETNUMVAL	(なし)								
GETPARTS	N		○	○	○	○			○
	Left, Top, Right, Bottom			○	○				○
MOVEPARTS	n		○	○	○	○			○
	x		○	○	○	○			○
	y		○	○	○	○			○
MOVEPOPW	n		○	○	○	○			○
	x		○	○	○	○			○
	y		○	○	○	○			○
MOVEPOPWDOWN	n		○	○	○	○			○
	y		○	○	○	○			○
MOVEPOPWLEFT	n		○	○	○	○			○
	x		○	○	○	○			○
MOVEPOPWRIGHT	n		○	○	○	○			○
	x		○	○	○	○			○
MOVEPOPWUP	n		○	○	○	○			○
	y		○	○	○	○			○
MSGBOX	S1			○	○		○		○
	S2			○	○		○		○
	S3		○	○	○	○			○
READCMEM	D	○		○	○				○
	S	○		○	○			○	○
	n					○			
RETURN	S		○	○	○	○			○
RSTALARMCNT	S		○	○	○	○			○
SHOWPAGE	S		○	○	○	○			○
SHOWPAGEBCD	S		○	○	○	○			○
STRCPY(W)	D			○	○				○
	S			○	○		○		○
STRM2W	D			○	○				○
	S			○	○		○		○
STRW2M	D			○	○				○
	S			○	○		○		○
WRITECMEM	D	○		○	○			○	○
	S	○		○	○				○
	n		○	○	○	○			
SETTIME	S			○					

NS シリーズ マクロリファレンス

関数名	引数	PT メモリ				定数	文字列	ホスト側 アドレス	INDEX 指定
		\$B \$HB \$SB	ビット変数 の数値 アクセス	\$W \$HW \$SW	ワード変数 のロング アクセス				
READCF	Mem			○					○
	Size			○					
	File			○			○		
	Dev			○		○			
WRITECF	Mem			○					○
	Size			○					
	File			○			○		
	Dev			○		○			
SWAP	S			○	○				○
	n			○	○	○			
SWAPL	S			○	○				
	n			○	○	○			
MEMCOPY	S			○	○				○
	D			○	○				○
	n			○					○
SETFOCUS	N					○			○
RELEASEFOCUS	(なし)								
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			○		○			○
ACOS	S			○		○			
ASIN	S			○		○			
ATAN	S			○		○			
ATAN2	S			○		○			
CEIL	S			○		○			

NS シリーズ マクロリファレンス

関数名	引数	PT メモリ				定数	文字列	ホスト側 アドレス	INDEX 指定
		\$B \$HB \$SB	ビット変数 の数値 アクセス	\$W \$HW \$SW	ワード変数 のロング アクセス				
COS	S			○		○			
DEG2RAD	S			○		○			
EXP	S			○		○			
FLOOR	S			○		○			
LOG	S			○		○			
LOG10	S			○		○			
RAD2DEG	S			○		○			
SQRT	S			○		○			
SIN	S			○		○			
TAN	S			○		○			
POW	S1,S2			○		○			
FADD	S1,S2			○		○			
FCOMP	S1,S2			○		○			
FDIV	S1,S2			○		○			
FMUL	S1,S2			○		○			
FSUB	S1,S2			○		○			
FSET	S			○		○			
LOCALTIME	S,D			○					

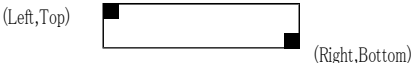
2-2 関数詳細

マクロプログラミングで使用する基本関数の詳細について説明します。

BCD	数値→BCD コード変換
書 式	BCD(S)
機 能	S の数値を BCD コードに変換します。 変換可能な範囲は、0～99999999 です。 この範囲を越える文字列を指定した場合は、桁落ちが発生します。 文字列の最後には 0 がセットされます。
リターン値	BCD コード
使用例	<pre>\$W0 = 1234; ‘\$W0 に数値 1234 をセット \$W10 = BCD(\$W0); ‘\$W10 に BCD コード(H1234) をセット \$W20L = 12345678; ‘\$W20～\$W21 に数値 12345678 をセット \$W22L = BCD(\$W20L); ‘\$W22～W23 に BCD コード(H12345678) をセット</pre>
BIN	BCD コード→数値変換
書 式	BIN(S)
機 能	BCD コード S を数値に変換します。 変換可能な BCD コードは H0～H99999999 です。
リターン値	数値
使用例	<pre>\$W0 = H1234; ‘\$W0 に BCD コード(H1234)をセット \$W10 = BIN(\$W0); ‘\$W10 に 1234 をセット \$W20L = H334455; ‘\$W20～\$W21 に BCD コード(H334455)をセット \$W22L = BIN(\$W20L); ‘\$W22～\$W23 に 334455 をセット</pre>

CLOSEPOPW	ポップアップウィンドウ画面を閉じる
書 式	CLOSEPOPW(n)
機 能	画面ページ番号nのポップアップ画面ウィンドウを閉じます。 n に設定可能な範囲は 0～3999 です。その他の値や存在しないポップアップ画面番号を設定した場合は無処理となります。
リターン値	なし
使用例	CLOSEPOPW(15); ‘画面ページ番号 15 のポップアップ画面ウィンドウを閉じる

GETNUMVAL	数値書込み値、変化値の取出し
書 式	GETNUMVAL ()
機 能	数値表示&入力部品の入力において書き込む値、または変化した値を取り出します。 数値表示&入力部品のマクロの実行条件が以下の時に使用できます。 システムバージョン V8.6 以下:「数値書込み前」、「数値変化時」 システムバージョン V8.7 以上:「数値入力開始前」、「数値書込み前」、「数値変化時」
リターン値	入力数値
使用例	\$W0=GETNUMVAL (); ‘\$W0 に数値書込みを行う値をセット

GETPARTS	部品の表示矩形取得
書 式	GETPARTS(n,Left,Top,Right,Bottom)
機 能	ID 番号がnの部品の表示矩形を取得します。矩形左上の座標 (Left,Top) と右下の座標 (Right,Bottom) がセットされます。  n に設定可能な範囲は 0～1023 です。その他の値や存在しない部品の ID 番号を設定した場合はリターン値-1 を返します。
リターン値	0 : 正常終了 -1 : 指定部品なし
使用例	GETPARTS(1,\$W0, \$W1, \$W2, \$W3); ‘(\$W0, \$W1)-(\$W2, \$W3)に ID 番号 1 の部品の表示矩形座標’をセット

MOVEPARTS	部品表示エリア移動
書 式	MOVEPARTS (n,x,y)
機 能	ID 番号が n の部品を指定座標 (x,y) へ移動します。 x,y には移動時の部品の左上の座標を指定してください。 n に設定可能な範囲は 0～1023 です。その他の値や存在しない部品の ID 番号を設定した場合はリターン値-1 を返します。 x,y に設定可能な値に制約はありませんが、部品が画面内に納まる範囲で設定してください。設定した値によっては、画面から部品の一部または全てが消える場合がありますので、注意してください。
リターン値	0 : 正常終了 -1 : 指定部品なし
使用例	MOVEPARTS (3,150,200); ‘ID 番号が 3 の部品を (150,200) の位置に移動

MEMO

- ◆ ON/OFF ボタン、ワードボタン、コマンドボタン、マルチファンクションのマクロに MOVEPARTS を設定し、その機能部品自体、あるいはその機能部品を含むフレームを移動させる場合は、「タッチ OFF 時」マクロを使用してください。「タッチ ON 時」マクロを使用すると、ボタンが押されたままになりますので、注意してください。

MOVEPOPW	ポップアップウィンドウ画面移動
書 式	MOVEPOPW(n,x,y)
機 能	画面ページ番号 n のポップアップ画面ウィンドウの左上を指定座標(x,y)へ移動します。 n に設定可能な範囲は 0～3999 です。その他の値や存在しないポップアップ画面番号を設定した場合はリターン値-1 を返します。 x,y に設定可能な値に制約はありませんが、ポップアップ画面が画面内に納まる範囲で設定してください。設定した値によっては、画面からポップアップ画面の一部または全てが消える場合がありますので、注意してください。
リターン値	0 : 正常終了 -1: 指定ページなし
使用例	\$W0=MOVEPOPW(10,140,160); ‘画面ページ番号 10 のポップアップ画面ウィンドウを(140, 160) ‘の位置へ移動後、\$W0 に 0 を返す。 ポップアップ画面が表示されていない場合は、\$W0 に-1 を返す。

MEMO

- ◆ ポップアップ画面上の ON/OFF ボタン、ワードボタン、コマンドボタン、マルチファンクションのマクロに MOVEPOPW を設定し、そのポップアップ画面を移動させる場合は、「タッチ OFF 時」マクロを使用してください。「タッチ ON 時」マクロを使用すると、ボタンが押されたままになりますので、注意してください。

MOVEPOPDOWN	ポップアップウィンドウ画面を下へ移動
書 式	MOVEPOPDOWN(n,y)
機 能	画面ページ番号 n のポップアップ画面ウィンドウを y ドット下へ移動します。 n に設定可能な範囲は 0～3999 です。その他の値や存在しないポップアップ画面番号を設定した場合はリターン値-1 を返します。 y に設定可能な値に制約はありませんが、ポップアップ画面が画面内に納まる範囲で設定してください。設定した値によっては、画面からポップアップ画面の一部または全てが消える場合がありますので、注意してください。
リターン値	0 : 正常終了 -1: 指定ページなし
使用例	\$W0=MOVEPOPDOWN(10,32); ‘画面ページ番号 10 のポップアップ画面ウィンドウを 32 ドット下へ移動後、\$W0 に 0 を返す。 ‘ポップアップ画面が表示されていない場合は、\$W0 に-1 を返す。

MEMO

- ◆ ポップアップ画面上の ON/OFF ボタン、ワードボタン、コマンドボタン、マルチファンクションのマクロに MOVEPOPDOWN を設定し、そのポップアップ画面を移動させる場合は、「タッチ OFF 時」マクロを使用してください。「タッチ ON 時」マクロを使用すると、ボタンが押されたままになりますので、注意してください。

MOVEPOPWLEFT	ポップアップウィンドウ画面を左へ移動
書 式	MOVEPOPWLEFT (n,x)
機 能	画面ページ番号 n のポップアップ画面ウィンドウを x ドット左へ移動します。 n に設定可能な範囲は 0～3999 です。その他の値や存在しないポップアップ画面番号を設定した場合はリターン値-1 を返します。 x に設定可能な値に制約はありませんが、ポップアップ画面が画面内に納まる範囲で設定してください。設定した値によっては、画面からポップアップ画面の一部または全てが消える場合がありますので、注意してください。
リターン値	0 : 正常終了 -1: 指定ページなし
使用例	\$W0=MOVEPOPWLEFT(10,32); ‘画面ページ番号 10 のポップアップ画面ウィンドウを 32 ドット左へ移動後、\$W0 に 0 を返す。 ‘ポップアップ画面が表示されていない場合は、\$W0 に-1 を返す。

MEN

- ◆ ポップアップ画面上の ON/OFF ボタン、ワードボタン、コマンドボタン、マルチファンクションのマクロに MOVEPOPWLEFT を設定し、そのポップアップ画面を移動させる場合は、「タッチ OFF 時」マクロを使用してください。「タッチ ON 時」マクロを使用すると、ボタンが押されたままになりますので、注意してください。

MOVEPOPWRIGHT	ポップアップウィンドウ画面を右へ移動
書 式	MOVEPOPWRIGHT(n,x)
機 能	画面ページ番号 n のポップアップ画面ウィンドウを x ドット右へ移動します。 n に設定可能な範囲は 0～3999 です。その他の値や存在しないポップアップ画面番号を設定した場合はリターン値-1 を返します。 x に設定可能な値に制約はありませんが、ポップアップ画面が画面内に納まる範囲で設定してください。設定した値によっては、画面からポップアップ画面の一部または全てが消える場合がありますので、注意してください。
リターン値	0 : 正常終了 -1: 指定ページなし
使用例	\$W0=MOVEPOPWRIGHT (10,32); ‘画面ページ番号 10 のポップアップ画面ウィンドウを 32 ドット右へ移動後、\$W0 に 0 を返す。 ‘ポップアップ画面が表示されていない場合は、\$W0 に-1 を返す。

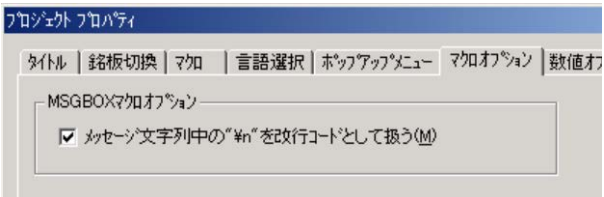
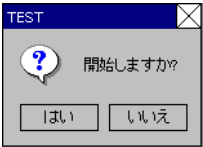
MEMO

- ◆ ポップアップ画面上の ON/OFF ボタン、ワードボタン、コマンドボタン、マルチファンクションのマクロに MOVEPOPWRIGHT を設定し、そのポップアップ画面を移動させる場合は、「タッチ OFF 時」マクロを使用してください。「タッチ ON 時」マクロを使用すると、ボタンが押されたままになりますので、注意してください。

MOVEPOPWUP	ポップアップウィンドウ画面を上へ移動
書 式	MOVEPOPWUP(n,y)
機 能	画面ページ番号 n のポップアップ画面ウィンドウを y ドット上へ移動します。 n に設定可能な範囲は 0～3999 です。その他の値や存在しないポップアップ画面番号を設定した場合はリターン値-1 を返します。 y に設定可能な値に制約はありませんが、ポップアップ画面が画面内に納まる範囲で設定してください。設定した値によっては、画面からポップアップ画面の一部または全てが消える場合がありますので、注意してください。
リターン値	0 : 正常終了 -1: 指定ページなし
使用例	\$W0=MOVEPOPWUP (10,32); ‘画面ページ番号 10 のポップアップ画面ウィンドウを 32 ドット上へ移動後、\$W0 に 0 を返す。 ‘ポップアップ画面が表示されていない場合は、\$W0 に-1 を返す。

MEMO

- ◆ ポップアップ画面上の ON/OFF ボタン、ワードボタン、コマンドボタン、マルチファンクションのマクロに MOVEPOPWUP を設定し、そのポップアップ画面を移動させる場合は、「タッチ OFF 時」マクロを使用してください。「タッチ ON 時」マクロを使用すると、ボタンが押されたままになりますので、注意してください。

MSGBOX	メッセージダイアログボックス表示								
書 式	MSGBOX(S1,S2,S3)								
機 能	指定したタイプのメッセージダイアログを表示します。 S1:メッセージ文字列 メッセージ中に改行を行う場合は、以下の手順を行ってください。 CX-Designer の[PT]-[プロジェクトプロパティ設定]からプロジェクトプロパティダイアログを表示します。[マクロオプション]タブの MSGBOX オプション-[メッセージ文字列中の”\n”を改行コードとして扱う]にチェックすると、メッセージ文字列内に”\n”を挿入することでメッセージを改行することができます。  S2:タイトルバー文字列 S3:メッセージダイアログで表示されるアイコンのタイプと応答ボタンのタイプを指定します。 4ビット(B0-B3) <table> <tr> <td>0:  STOP マーク</td> <td>1:  QUESTION マーク</td> </tr> <tr> <td>2:  EXCLAMATION マーク</td> <td>3:  INFORMATION マーク</td> </tr> </table> 4ビット(B4-B7) 0:[OK]ボタン 1:[OK]/[キャンセル]ボタン 2:[再試行]/[キャンセル]ボタン 3:[はい]/[いいえ]ボタン 4:[はい]/[いいえ]/[キャンセル]ボタン 5:[中止]/[再試行]/[無視]ボタン	0:  STOP マーク	1:  QUESTION マーク	2:  EXCLAMATION マーク	3:  INFORMATION マーク				
0:  STOP マーク	1:  QUESTION マーク								
2:  EXCLAMATION マーク	3:  INFORMATION マーク								
リターン値	<table> <tr> <td>0:[OK]ボタンを選択</td> <td>1:[キャンセル]ボタンを選択</td> </tr> <tr> <td>2:[はい]ボタンを選択</td> <td>3:[いいえ]ボタンを選択</td> </tr> <tr> <td>4:[無視]ボタンを選択</td> <td>5:[再試行]ボタンを選択</td> </tr> <tr> <td>6:[中止]ボタンを選択</td> <td></td> </tr> </table>	0:[OK]ボタンを選択	1:[キャンセル]ボタンを選択	2:[はい]ボタンを選択	3:[いいえ]ボタンを選択	4:[無視]ボタンを選択	5:[再試行]ボタンを選択	6:[中止]ボタンを選択	
0:[OK]ボタンを選択	1:[キャンセル]ボタンを選択								
2:[はい]ボタンを選択	3:[いいえ]ボタンを選択								
4:[無視]ボタンを選択	5:[再試行]ボタンを選択								
6:[中止]ボタンを選択									
使用例	<pre> \$W0=MSGBOX(“開始しますか?”,“TEST”,H31); ‘H31:[はい]/[いいえ]ボタン、QUESTION マーク表示 IF(\$W0==2) ‘はいを選択したときの処理を記述 ELSE ‘いいえを選択したときの処理を記述 ENDIF </pre> 上記設定の場合、以下のようなメッセージダイアログが表示されます。 								

MEMO

- ◆ MSGBOX により同時に表示することのできるメッセージボックスは 1 つまでです。
既に 1 つ表示されている状態で MSGBOX を実行した場合、メッセージボックスは表示されず、リターン値として 1 が返ります。

例 次の 2 つのビットランプを作成した場合

	通信アドレス	マクロ
ランプ 1	\$B0	\$W0=MSGBOX(“メッセージ 1”, “タイトル 1”, H31);
ランプ 2	\$B0	\$W1=MSGBOX(“メッセージ 2”, “タイトル 2”, H31);

仮にランプ 1 のマクロが先に実行されるとします。\$B0 の値を変化すると、ランプ 1 のメッセージボックスが表示されます。ランプ 2 のメッセージボックスは表示されず、\$W1 に 1 が格納されます。

- ◆ システムメニュー[PT 動作設定]タブで[ブザー音]を[ON]または[ERR ON]と設定している場合、アイコンに STOP、EXCLAMATION を指定した時、メッセージダイアログ表示中にブザーが鳴ります。

READCMEM	指定通信アドレスからのデータ読出し				
書 式	READCMEM(D, S, n)				
機 能	S で指定したホストの通信アドレスから n ビット/n チャネル分読み出し、D にコピーします。読出し可能な最大点数は以下ようになります。 <table border="1"> <tr> <td>ビット</td><td>126 ビット</td></tr> <tr> <td>ワード</td><td>126 チャネル</td></tr> </table> n に範囲外の値を設定した場合、通信エラーまたはマクロ実行エラーになります。読み出し可能な最大点数は、接続する PLC 機種によって異なります。	ビット	126 ビット	ワード	126 チャネル
ビット	126 ビット				
ワード	126 チャネル				
リターン値	なし				
使用例	READCMEM(\$W0,[HOST1:DM0],10); ‘ホスト名 HOST1 と設定されている PLC の DM0～DM9 を’ ‘\$W0～\$W9 に読み出します。				

MEMO

- ◆ S にタグを指定した場合は、n にはタグのデータ型に従い、下表の値を設定してください。

タグのデータ型	n に設定可能な値
BOOL	1
UINT、INT、UINT_BCD、CHANNEL、WORD	1
UDINT、DINT、UDINT_BCD、REAL、DWORD	2
ULINT、LINTU、LINT_BCD、LREAL、LWORD	4
STRING	文字数以下(*1)
配列	要素数以下(*1)

*1:126 を超えない範囲で設定してください。

例:[HOST1:TAG1]のデータ型が DINT の場合

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

NS シリーズ マクロリファレンス

RETURN	マクロプログラムの終了
書 式	RETURN(S)
機 能	S の値が 0 のときマクロプログラムを終了し、機能部品の処理を続行します。0 以外 のときマクロプログラムを終了し、機能部品の処理も中断します。
リターン値	なし
使用例	RETURN(0); ‘マクロを終了し、機能部品の処理を続行 RETURN(1); ‘マクロを終了し、機能部品の処理を中断

RSTALARMCNT	アラーム／イベント発生回数を 0 クリア
書 式	RSTALARMCNT(S)
機 能	S の値が 0 のときアラーム発生回数を、1 のときイベント発生回数をそれぞれクリアしま す。
リターン値	0 : 正常終了 -1: 対応機能なし
使用例	RSTALARMCNT(0); ‘アラーム発生回数をクリア RSTALARMCNT(1); ‘イベント発生回数をクリア

SHOWPAGE	画面切換
書 式	SHOWPAGE(S)
機 能	S で指定した画面ページ番号の画面へ切り換えます。 S に設定可能な範囲は 0～3999 です。範囲外の値を設定した場合は、マクロ実行時エラーとなり、また存在しない画面番号を設定した場合は、ページ読出しエラーとなります。
リターン値	なし
使用例	SHOWPAGE(10); ‘画面ページ番号 10 へ画面を切換え

MEMO

- ◆ SHOWPAGE 以降に書かれたマクロは実行されません。SHOWPAGE は最後の行に記述してください。

悪い例

SHOWPAGE(10); ←画面 10 に切り換わります。
\$W50=100; ←\$W50 への代入は実行されません。

良い例

\$W50=100; ←\$W50 に 100 が代入されます。
SHOWPAGE(10); ←画面 10 に切り換わります。

同様に、マルチファンクションに設定したマクロで SHOWPAGE を実行する場合、マクロ機能以降に設定された機能は実行されません。SHOWPAGE により画面切換を行なう場合は、最後に実行する機能としてマクロを設定してください。

NS シリーズ マクロリファレンス

STRCPY(W)	文字列コピー
書 式	STRCPY(D, S) マルチバイトコード STRCPYW(D, S) ユニコード(UTF-16)
機 能	S から D へ文字列をコピーします。 コピーは、null まで行います。(null を含みます)
リターン値	なし
使用例	STRCPYW(\$W0, "ABC"); '\$W0～\$W2 に"ABC"をセット \$W100=H6400; STRCPY(\$W110,\$W100); '\$W110 に"d"をセット

MEMO

- ◆ 'null'はマルチバイトコード(S-JIS)では'00'、ユニコードでは'0000'に相当します。
- ◆ コピーには null を含むため、\$W32767 に文字をセットする場合は注意してください。STRCPY(W)実行時、null までコピーされるため、文字データ+null が\$W32767 を越えないよう注意して下さい。(\$W32767 を越える場合は通信エラーが発生します)

STRM2W	マルチバイト文字列をユニコード(UTF-16)文字列に変換
書 式	STRM2W(D,S)
機 能	Sで指定したマルチバイトの文字列をユニコードに変換して D にコピーします。 コピーは null まで行います。(null を含みます)
リターン値	なし
使用例	STRM2W(\$W0,"ABC"); "ABC"をユニコードに変換して\$W0～\$W2 にコピー

STRW2M	ユニコード(UTF-16)文字列をマルチバイト文字列に変換
書 式	STRW2M(D,S)
機 能	Sで指定したユニコードの文字列をマルチバイトに変換して D にコピーします。 コピーは null まで行います。(null を含みます)
リターン値	なし
使用例	STRW2M(\$W0, "ABC"); “ABC”をマルチバイトに変換して\$W0～\$W1 にコピー

WRITECMEM	指定通信アドレスへのデータ書込み				
書 式	WRITECMEM(D,S,n)				
機 能	Sからnビット/n チャンネル分のデータを、D で指定したホストの通信アドレスへコピーします。書込み可能な最大点数は以下ようになります。 <table border="1"> <tr> <td>ビット</td><td>126 ビット</td></tr> <tr> <td>ワード</td><td>126 チャンネル</td></tr> </table> n に範囲外の値を設定した場合、通信エラーまたは、マクロ実行エラーになります。書込み可能な最大点数は、接続するPLC機種によって異なります。	ビット	126 ビット	ワード	126 チャンネル
ビット	126 ビット				
ワード	126 チャンネル				
リターン値	なし				
使用例	WRITECMEM([HOST1:DM0],\$W0,10); ‘\$W0～\$W9 の値をホスト名 HOST1 と設定されている PLC の ‘DM0～DM9 に書き込みます。				

MEMO

- ◆ D にタグを指定した場合は、n にはタグのデータ型に従い、下表の値を設定してください。

タグのデータ型	n に設定可能な値
BOOL	1
UINT、INT、UINT_BCD、CHANNEL、WORD	1
UDINT、DINT、UDINT_BCD、REAL、DWORD	2
ULINT、LINTU、LINT_BCD、LREAL、LWORD	4
STRING	文字数以下(*1)
配列	要素数以下(*1)

*1:126 を超えない範囲で設定してください。

例:[HOST1:TAG1]のデータ型が DINT の場合

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

追加マクロ関数詳細

NS シリーズ Ver.3 より追加された、マクロの関数詳細は以下のとおりです。

マクロ名	SETTIME												
書式	SETTIME(S)												
機能	指定内部メモリアドレスから年月日時分秒を指定して、NS 本体の内部時計を変更します。S は内部メモリアドレス\$W(\$W0～\$W32765)、または内部保持メモリアドレス\$HW(\$HW0～\$HW8189)を指定します(*1)。指定内部メモリアドレスにはBCD または BIN 形式で日時データを設定します(*2)。また、指定内部メモリアドレスより 3CH 分のデータを設定データとして使用します(*3)。 *1 アドレス指定ではインデックス指定も可能です。 *2 下記 MEMO 欄参照 *3 3CH 分の日時データは、下記のように格納されている必要があります。 <table><tr><th>アドレス</th><th>上位バイト(ビット08～15)</th><th>下位バイト(ビット00～07)</th></tr><tr><td>1CH</td><td>分(00～59)</td><td>秒(00～59)</td></tr><tr><td>2CH</td><td>日(01～31)</td><td>時(00～23)</td></tr><tr><td>3CH</td><td>月(01～12)</td><td>年(00～99)</td></tr></table>	アドレス	上位バイト(ビット08～15)	下位バイト(ビット00～07)	1CH	分(00～59)	秒(00～59)	2CH	日(01～31)	時(00～23)	3CH	月(01～12)	年(00～99)
アドレス	上位バイト(ビット08～15)	下位バイト(ビット00～07)											
1CH	分(00～59)	秒(00～59)											
2CH	日(01～31)	時(00～23)											
3CH	月(01～12)	年(00～99)											
リターン値	なし												
使用例	・PLC の内部時計データ(BCD 形式)を読み出し、NS 本体の内部時計に設定する場合 'PLC(CS/CJ/CP シリーズ)時計データ:AR351～AR353 の読出 READCMEM(\$W1000,[SERIALA:AR351],3); SETTIME(\$W1000); ・BIN 形式で『2002 年 12 月 31 日 18 時 59 分 30 秒』を設定する場合 \$W100=BIN(H30); '秒データ \$W101=BIN(H59); SWAP(\$W101,1); '分データ \$W102=BIN(H18); '時データ \$W103=BIN(H31); SWAP(\$W103,1); '日データ \$W104=BIN(H12); '月データ \$W105=BIN(H02); SWAP(\$W105,1); '年データ \$W200=\$W100+\$W101; \$W201=\$W102+\$W103; \$W202=\$W104+\$W105; SETTIME(\$W200);												

MEMO

- ◆ 引数 S (日付データ) は、システムメモリ\$SW14～\$SW17(日付・時間)の格納形式と合わせる必要があります。データ形式が異なる場合、下表に示すように SETTIME は実行されません。

		SETTIME の引数 S	
		BCD	バイナリ
\$SW14～17 格納形式	BCD	○	×
	バイナリ	×	○

○：SETTIME 実行可能

×：[マクロプログラム実行エラー]が表示されます。

SETTIME の引数 S と格納形式が同じデータ形式となるように設定して下さい。

システムメモリの格納形式は、[システム設定]-[イニシャル]タブから[システムメモリー一覧]で設定できます。

格納形式(デフォルト:BCD)

※チェックの有無は関係ありません。

NS シリーズ マクロリファレンス

NS シリーズ Ver.4 より追加された、マクロの関数詳細は以下のとおりです。

マクロ名	WRITECF
書式	WRITECF(Mem,Size,File,Dev)
機能	PT メモリの内容をメモ리카ード (CF) 上の指定ファイルに保存します。ファイルには指定位置の PT メモリの内容が、そのままバイナリ値で書き込まれます。 Mem: 保存元データの先頭 PT メモリ番号 (\$W・\$HW・\$SW) インデックス指定も可能です。\$W に設定可能な範囲は 0～32767 です。\$HW に設定可能な範囲は 0～8191 です。 \$B/\$HB/\$SB を指定することはできません。 Size: 保存サイズ (単位: ワード) LONG 値直接または \$W・\$HW での間接指定 (2 ワード使用) ができます。 \$W に設定可能な範囲は 0～32767 です。\$HW に設定可能な範囲は 0～8191 です。 設定範囲を越えた場合はアクセスエラーとなり、エラーメッセージが表示されます。 File: 保存先のファイル名 文字列直接または \$W・\$HW での間接指定が可能です。File で指定したファイルが保存先にすでに存在する場合は、確認メッセージは表示せずに上書きで保存されます。(リターン値は正常終了です。) ファイル名には、半角英数 (“0”～“9”、“A”～“Z”、“a”～“z”、“\$”、“_”) で拡張子を含め 43 文字まで指定できます。 Dev: 書き込み先はメモ리카ードのみになる為、0 を指定してください。
リターン値	0: 正常終了 -1: 保存失敗
使用例	① \$W100 = WRITECF(\$W1000, 128, “CF_FILE.BIN” , 0); ② \$W2000L=128; STRCPY(\$W2002, “CF_FILE.BIN”); \$W100 = WRITECF(\$W1000, \$W2000, \$W2002 , 0);

NS シリーズ マクロリファレンス

マクロ名	READCF
書式	READCF (Mem,Size,File,Dev)
機能	メモ리카ード (CF) 上の指定ファイルの内容 (バイナリ値) を PT メモリに読み出します。 Mem: 読出先の先頭 PT メモリ番号 (\$W・\$HW・\$SW) インデックス指定も可能です。\$W に設定可能な範囲は 0～32767 です。 \$HW に設定可能な範囲は 0～8191 です。 \$B/\$HB/\$SB を指定することはできません。 Size: 読出サイズ (単位: ワード) LONG 値直接または \$W・\$HW での間接指定 (2 ワード使用) が可能。 \$W に設定可能な範囲は 0～32767 です。\$HW に設定可能な範囲は 0～8191 です。 設定範囲を越えた場合はアクセスエラーとなり、エラーメッセージが表示されます。 Size に 0 以下の値を指定した場合、ファイルサイズ分が PT メモリに読み出されます。 Size (Size>0) よりもファイルサイズが大きい場合は、Size 分のみ読み出されます。(リターン値は正常終了です。) Size (Size>0) よりもファイルサイズが小さい場合は、ファイルサイズ分のみ読み出しが実行されます。(リターン値は正常終了です。) File: 読出元のファイル名 文字列直接または \$W・\$HW での間接指定が可能です。読み出しはワード単位で実行され、ファイルサイズが奇数バイトである場合、最後の 1 バイトは PT メモリに読み出されません。(リターン値は正常終了です。) ファイル名には、半角英数 (“0”～“9”、“A”～“Z”、“a”～“z”、“\$”、“_”) で拡張子を含め 43 文字まで指定できます。 Dev: 読み出し先のデバイスを指定します。読み出し先はメモ리카ードのみになるため、0 を指定してください。
リターン値	0: 正常終了 -1: 読出失敗
使用例	① \$W100 = READCF(\$W1000, 0, “CF_FILE.BIN” , 0); ② \$W2000L=0; STRCPY(\$W2002, “CF_FILE.BIN”); \$W100 = READCF(\$W1000, \$W2000, \$W2002 , 0);

マクロ名	SWAP
書式	SWAP(S,n)
機能	Sを先頭とする\$W・\$HWからnワード分の上位バイト、下位バイトのデータを入れ替えます。 S: 入れ替える\$W・\$HW データの先頭アドレス *1 n: 入れ替えるワード数 *1 直接指定の場合 1～32767 間接指定の場合 \$W0～\$W32767 \$HW0～\$HW8191 *1 アドレス指定ではインデックス指定も可能です。\$W に設定可能な範囲は 0～32767 です。\$HW に設定可能な範囲は 0～8191 です。
リターン値	なし
使用例	使用例: SWAP(\$W100, 3); ... \$W100 から 3 ワード分の上位バイト、下位バイトを入れ替え。 <SWAP 変換前> 上位バイトと下位バイトを入れ替え <SWAP 変換後>

NS シリーズ マクロリファレンス

マクロ名	SWAPL																																																												
書式	SWAPL (S,n)																																																												
機能	S を先頭とする\$W・\$HW から n ロングワード分の上位バイト、下位バイトをロングワード単位で入れ替えます。 S: 入れ替える\$W・\$HW データの先頭アドレス *1 n: 入れ替えるワード数 *1 直接指定の場合 1～16384 間接指定の場合 \$W0～\$W32767 \$HW0～\$HW8191 *1 アドレス指定ではインデックス指定も可能です。\$W に設定可能な範囲は 0～32767 です。\$HW に設定可能な範囲は 0～8191 です。																																																												
リターン値	なし																																																												
使用例	SWAPL(\$100,3);…\$W100 から 3 ロングワード分の上位バイト、下位バイトを入れ替え。 <SWAPL 変換前> <table><tr><td>\$W100</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>\$W101</td><td>5</td><td>6</td><td>7</td><td>8</td></tr><tr><td>\$W102</td><td>9</td><td>0</td><td>1</td><td>2</td></tr><tr><td>\$W103</td><td>3</td><td>4</td><td>5</td><td>6</td></tr><tr><td>\$W104</td><td>7</td><td>8</td><td>9</td><td>0</td></tr><tr><td>\$W105</td><td>1</td><td>2</td><td>3</td><td>4</td></tr></table> ← ロングワ ード単位 で上位と 下位を 入れ替 え → <SWAPL 変換後> <table><tr><td>\$W100</td><td>5</td><td>6</td><td>7</td><td>8</td></tr><tr><td>\$W101</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>\$W102</td><td>3</td><td>4</td><td>5</td><td>6</td></tr><tr><td>\$W103</td><td>9</td><td>0</td><td>1</td><td>2</td></tr><tr><td>\$W104</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>\$W105</td><td>7</td><td>8</td><td>9</td><td>0</td></tr></table>	\$W100	1	2	3	4	\$W101	5	6	7	8	\$W102	9	0	1	2	\$W103	3	4	5	6	\$W104	7	8	9	0	\$W105	1	2	3	4	\$W100	5	6	7	8	\$W101	1	2	3	4	\$W102	3	4	5	6	\$W103	9	0	1	2	\$W104	1	2	3	4	\$W105	7	8	9	0
\$W100	1	2	3	4																																																									
\$W101	5	6	7	8																																																									
\$W102	9	0	1	2																																																									
\$W103	3	4	5	6																																																									
\$W104	7	8	9	0																																																									
\$W105	1	2	3	4																																																									
\$W100	5	6	7	8																																																									
\$W101	1	2	3	4																																																									
\$W102	3	4	5	6																																																									
\$W103	9	0	1	2																																																									
\$W104	1	2	3	4																																																									
\$W105	7	8	9	0																																																									

マクロ名	MEMCOPY																																																												
書式	MEMCOPY(S, D, n);																																																												
機能	PT メモリ\$W・\$HW の内容をコピーします。 S:コピー元の\$W・\$HW データの先頭アドレス *1 D:コピー先の\$W・\$HW データの先頭アドレス *1 N:コピーするワード数 *1 直接指定の場合 1～32767 間接指定の場合 \$W0～\$W32767 \$HW0～\$HW8191 *1 アドレス指定ではインデックス指定も可能です。\$W に設定可能な範囲は 0～32767 です。\$HW に設定可能な範囲は 0～8191 です。																																																												
リターン値	なし																																																												
使用例	MEMCOPY(\$W100,\$W102,2); \$W100 を先頭とするアドレス 2 ワード分\$W102 を先頭とするエリアにコピーする。 <MEMCOPY 前> <table><tr><td>\$W100</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>\$W101</td><td>5</td><td>6</td><td>7</td><td>8</td></tr><tr><td>\$W102</td><td>9</td><td>0</td><td>1</td><td>2</td></tr><tr><td>\$W103</td><td>3</td><td>4</td><td>5</td><td>6</td></tr><tr><td>\$W104</td><td></td><td></td><td></td><td></td></tr><tr><td>\$W105</td><td></td><td></td><td></td><td></td></tr></table> n ワード分 → <MEMCOPY 後> <table><tr><td>\$W100</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>\$W101</td><td>5</td><td>6</td><td>7</td><td>8</td></tr><tr><td>\$W102</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>\$W103</td><td>5</td><td>6</td><td>7</td><td>8</td></tr><tr><td>\$W104</td><td></td><td></td><td></td><td></td></tr><tr><td>\$W105</td><td></td><td></td><td></td><td></td></tr></table>	\$W100	1	2	3	4	\$W101	5	6	7	8	\$W102	9	0	1	2	\$W103	3	4	5	6	\$W104					\$W105					\$W100	1	2	3	4	\$W101	5	6	7	8	\$W102	1	2	3	4	\$W103	5	6	7	8	\$W104					\$W105				
\$W100	1	2	3	4																																																									
\$W101	5	6	7	8																																																									
\$W102	9	0	1	2																																																									
\$W103	3	4	5	6																																																									
\$W104																																																													
\$W105																																																													
\$W100	1	2	3	4																																																									
\$W101	5	6	7	8																																																									
\$W102	1	2	3	4																																																									
\$W103	5	6	7	8																																																									
\$W104																																																													
\$W105																																																													

NS シリーズ Ver.5 より追加された、マクロの関数詳細は以下のとおりです。

マクロ名	SETFOCUS
書式	SETFOCUS(n);
機能	指定 ID 番号の数値表示&入力部品、または文字列表示&入力部品にフォーカスをセットします。 n:フォーカスをセットする部品 ID 番号(0～32767) 入力順序の先頭に設定されている部品にフォーカスをセットする場合は、-1 ●すでに何かの部品にフォーカスがセットされている場合は、本マクロ命令は動作しません。 ●指定 ID 番号の部品が画面上に表示されていないフレームページに配置されている場合は、フォーカスはセットされません。(実行時、「マクロプログラム実行エラー」が表示されます。) ●数値表示&入力部品・文字列表示&入力部品以外の部品 ID 番号を指定したとき、本マクロ命令は動作しません。 ●シート上の部品にフォーカスをセットすることはできません。 SETFOCUS は以下のマクロ実行タイミングでは動作しません。 ープロジェクトロード時マクロ ー画面ロード時マクロ ー画面アンロード時マクロ ー数値表示&入力部品の数値入力開始前マクロ ー数値表示&入力部品の数値書込み前マクロ ー文字列表示&入力部品の文字列入力開始前マクロ ー文字列表示&入力部品の文字列書込み前マクロ
リターン値	0=正常終了
	-1=指定した ID の部品が見つからない。
使用例	部品 ID 番号 4 の部品にフォーカスをセットする場合 SETFOCUS(4);

NS シリーズ マクロリファレンス

マクロ名	RELEASEFOCUS
書式	RELEASEFOCUS(); 引数なし
機能	画面上の数値表示&入力部品または文字列表示&入力部品にフォーカスがセットされている場合、フォーカスを解除します。 ● 表示中の画面上にフォーカスがセットされている部品がある場合、フォーカスを解除します。 ● フォーカスがセットされている部品がない場合は、本マクロ命令は動作しません。 ● 表示中のフレームページに配置されている部品にフォーカスがセットされている場合も、フォーカスを解除します。 ● シート上の部品にフォーカスがセットされている場合も、フォーカスを解除します。 RELEASEFOCUS は次のマクロ実行タイミングでは動作しません。 - ープロジェクトロード時マクロ - ー画面ロード時マクロ - ー画面アンロード時マクロ - ー数値表示&入力部品の数値入力開始前マクロ - ー数値表示&入力部品の数値書込み前マクロ - ー文字列表示&入力部品の文字列入力開始前マクロ - ー文字列表示&入力部品の文字列書込み前マクロ
リターン値	なし
使用例	RELEASEFOCUS();

NS シリーズ マクロリファレンス

NS シリーズ Ver.6 より以下のマクロ演算機能が追加されました。

マクロ名	SHOWPAGEBCD
書式	SHOWPAGEBCD(S)
機能	S:画面ページ番号(H0～H3999) S で指定した画面ページ番号の画面へ切り換えます。 S に設定可能な範囲は H0～H3999 です。範囲外の値を設定した場合、または BCD として不正な値が指定された場合は、マクロ実行時エラーとなります。また、存在しない画面番号を設定した場合は、ページ読出しエラーとなります。
リターン値	なし
使用例	- 画面ページ番号 10 へ画面を切り換える SHOWPAGEBCD(H10); - 画面ページ番号を間接指定し、画面ページ番号 10 へ画面を切り換える \$W100 = H10; SHOWPAGEBCD(\$W100);

MEMO

- ◆ SHOWPAGEBCD 以降に書かれたマクロは実行されません。SHOWPAGEBCD は最後の行に記述してください。

悪い例

SHOWPAGEBCD(H10); ←画面 10 に切り換わります。
\$W50=100; ←\$W50 への代入は実行されません。

良い例

\$W50=100; ←\$W50 に 100 が代入されます。
SHOWPAGEBCD(H10); ←画面 10 に切り換わります。

同様に、マルチファンクションに設定したマクロで SHOWPAGEBCD を実行する場合、マクロ機能以降に設定された機能は実行されません。SHOWPAGEBCD により画面切替を行なう場合は、最後に実行する機能としてマクロを設定してください。

NS シリーズ マクロリファレンス

マクロ名	FOR,NEXT
書式	FOR(n),NEXT
機能	n:ループ数 FOR から NEXT 間のプログラムを指定回数繰り返します。入れ子(ネスト)は1つまでです。したがって、FOR～NEXT の中に FOR～NEXT を記述することはできません。 *nのループ数に指定できる範囲は 0～32767 です。負の値が設定された場合は 0 とみなします。 指定可能なアドレスは\$W, \$HW, \$SW です。
リターン値	なし
使用例	‘FOR～NEXT を 10 回繰り返して実行し、\$W0～\$W9 へ初期値 0 を代入する。 <pre> \$W0=0; \$SW27=0; FOR(10) \$W0I0=0; \$SW27=\$SW27+1; NEXT; </pre>

マクロ名	BREAK
書式	BREAK;
機能	FOR から NEXT 間のプログラムの繰り返しを中断します。
リターン値	なし
使用例	’\$W100I0>30 が成立すれば、FOR ループを抜け出す <pre> \$SW27=0; FOR(10) \$W100I0=\$W50I0+10; IF(\$W100I0>30) BREAK; ENDIF \$SW27=\$SW27+1; NEXT; </pre>

マクロ名	CONTINUE
書式	CONTINUE;
機能	FOR から NEXT 間のプログラムにおいて、FOR ループの先頭に戻り、次の繰り返し処理に入ります
リターン値	なし
使用例	<pre>'\$W50I0>30 が成立すれば、FOR ループの先頭に戻り、次の繰り返し処理に入る \$SW27=0; FOR(10) IF(\$W50I0>30) \$SW27=\$SW27+1; CONTINUE; ENDIF \$W100I0=\$W50I0+10; \$SW27=\$SW27+1; NEXT;</pre>

NS シリーズ Ver.6.2 より追加された、マクロの関数詳細は以下のとおりです。

マクロ名	READHOSTB
書式	READHOSTB (D, h, ch, addr, r, n)
機能	ホスト h のビットデータを n ビット分読み出し、PT メモリ(\$B/\$HB)D にコピーします。 D: 読込先先頭アドレス(\$B0~\$B32767, \$HB0~\$HB8191) h: ホスト(ホスト名/ホスト ID) ch: ホストアドレス種別 *1 addr: ホスト先頭アドレス r: リレー番号 n: 書込要素数(1~126) *1: 章末のアドレス種別番号を参照ください。
リターン値	正常終了: 0x0000 異常発生時: 上位 8 ビット(B8~B15): MRES (メインレスポンスコード) 下位 8 ビット(B0~B7): SRES (サブレスポンスコード) MRES、SRES については NS シリーズプログラミングマニュアル「5-2-7 通信エラーとその対処」を参照ください。
使用例	ホストに SerialA、SerialB が登録されている場合 - ホスト ID1 (シリアルポート A) に接続されている PLC の CIO1000.00 から 10 ビットのデータを読み出し、\$B10~\$B19 に格納 READHOSTB (\$B10, 1, 100, 1000, 0, 10); - ホスト名称=[SerialB] (シリアルポート B) に接続されている PLC の DM2000.05 から 10 ビットのデータを読み出し、\$HB10~\$HB19 に格納 READHOSTB (\$HB10, [SerialB], 300, 2000, 5, 10);

MEMO

- ◆ CX-Designer でホストを削除した場合、削除したホスト以降のホスト ID が前詰めされます。下表の例では、HOST2 を削除すると、ホスト ID が前詰めされ、HOST3 のホスト ID が 2 に変更されます。

ホスト削除前		ホスト削除後	
ホスト ID	ホスト名	ホスト ID	ホスト名
1	SERIALA	1	SERIALA
2	HOST2	2	HOST3
3	HOST3	—	—

READHOSTB の引数 h としてホスト ID を使用している場合は、変更後のホスト ID を指定し直す必要がありますので、注意してください。上記の例の場合、次のようにマクロを変更します。

ホスト削除前のマクロ	READHOSTB(\$B10, <u>3</u> ,100,1000,0,10);	HOST3 を指定するための 第二引数の「3」を「2」に変更
ホスト削除後のマクロ	READHOSTB(\$B10, <u>2</u> ,100,1000,0,10);	

ホスト ID は CX-Designer の[通信設定]ダイアログでホスト名の左に表示される番号を確認してください。

- ◆ CX-Designer でホスト名を変更したとき、READHOSTB の引数 h としてホスト名称を使用している場合には、変更後のホスト名に指定し直す必要がありますので、注意してください。

例えば、ホスト名を HOST_1 から HOST_A に変更した場合は、次のように変更します。

ホスト名変更前のマクロ	READHOSTB(\$B10,[HOST_1],100,1000,0,10);	[HOST_1]を
ホスト名変更後のマクロ	READHOSTB(\$B10,[HOST_A],100,1000,0,10);	[HOST_A]に変更

NS シリーズ マクロリファレンス

マクロ名	READHOSTW
書式	READHOSTW (D, h, ch, addr, n)
機能	ホスト h のワードデータを n ワード分読み出し、PT メモリ(\$W/\$HW)D にコピーします。 D: 読込先先頭アドレス(\$W0～\$W32767、\$HW0～\$HW8191) h: ホスト(ホスト名/ホスト ID) ch: ホストアドレス種別 *1 addr: ホスト先頭アドレス n: 書込要素数(1～126) *1: 章末のアドレス種別番号を参照ください。
リターン値	正常終了: 0x0000 異常発生時: 上位 8 ビット(B8～B15): MRES (メインレスポンスコード) 下位 8 ビット(B0～B7): SRES (サブレスポンスコード) MRES、SRES については NS シリーズプログラミングマニュアル「5-2-7 通信エラーとその対処」を参照ください。
使用例	ホストに SerialA、SerialB が登録されている場合 - ホスト ID1 (シリアルポート A) に接続されている PLC の CIO1000 から 10 ワードのデータを読み出し、\$W10～\$W19 に格納 READHOSTW (\$W10, 1, 100, 1000, 10); - ホスト名称=[SerialB] (シリアルポート B) に接続されている PLC の DM2000 から 10 ワードのデータを読み出し、\$HW10～\$HW19 に格納 READHOSTW (\$HW10, [SerialB], 300, 2000, 10);

NEW

- ◆ CX-Designer でホストを削除した場合、削除したホスト以降のホスト ID が前詰めされます。下表の例では、HOST2 を削除すると、ホスト ID が前詰めされ、HOST3 のホスト ID が 2 に変更されます。

ホスト削除前		ホスト削除後	
ホスト ID	ホスト名	ホスト ID	ホスト名
1	SERIALA	1	SERIALA
2	HOST2	2	HOST3
3	HOST3	—	—

READHOSTW の引数 h としてホスト ID を使用している場合は、変更後のホスト ID を指定し直す必要がありますので、注意してください。上記の例の場合、次のようにマクロを変更します。

ホスト削除前のマクロ	READHOSTW(\$W10, <u>3</u> ,100,1000,10);	HOST3 を指定するための 第二引数の「3」を「2」に変更
ホスト削除後のマクロ	READHOSTW(\$W10, <u>2</u> ,100,1000,10);	

ホスト ID は CX-Designer の[通信設定]ダイアログでホスト名の左に表示される番号を確認してください。

- ◆ CX-Designer でホスト名を変更したとき、READHOSTW の引数 h としてホスト名称を使用している場合には、変更後のホスト名に指定し直す必要がありますので、注意してください。

例えば、ホスト名を HOST_1 から HOST_A に変更した場合は、次のように変更します。

ホスト名変更前のマクロ	READHOSTW(\$W10, <u>[HOST_1]</u> ,100,1000,10);	[HOST_1]を
ホスト名変更後のマクロ	READHOSTW(\$W10, <u>[HOST_A]</u> ,100,1000,10);	[HOST_A]に変更

NS シリーズ マクロリファレンス

マクロ名	WRITEHOSTB
書式	WRITEHOSTB (h, ch, addr, r, S, n)
機能	PT メモリ(\$B/\$HB)S から n ビット分のビットデータを、指定したホスト h にコピーします。 h:ホスト(ホスト名/ホスト ID) ch:ホストアドレス種別 *1 addr:ホスト先頭アドレス r:リレー番号 S:書込元先頭アドレス(\$B0~\$B32767, \$HB0~\$HB8191) n:書込要素数(1~126) *1:章末のアドレス種別番号を参照ください。
リターン値	正常終了:0x0000 異常発生時:上位 8 ビット(B8~B15):MRES(メインレスポンスコード) 下位 8 ビット(B0~B7):SRES(サブレスポンスコード) MRES、SRES については NS シリーズプログラミングマニュアル「5-2-7 通信エラーとその対処」を参照ください。
使用例	ホストに SerialA、SerialB が登録されている場合 - ホスト ID1 (シリアルポート A) に接続されている PLC の CIO1000.00 に \$B10~\$B19 のデータを書き込む WRITEHOSTB (1, 100, 1000, 0, \$B10, 10); - ホスト名称=[SerialB] (シリアルポート B) に接続されている PLC の DM1000.05 に \$HB10~\$HB19 のデータを書き込む WRITEHOSTB ([SerialB], 300, 1000, 5, \$HB10, 10);

MEMO

- ◆ CX-Designer でホストを削除した場合、削除したホスト以降のホスト ID が前詰めされます。
下表の例では、HOST2 を削除すると、ホスト ID が前詰めされ、HOST3 のホスト ID が 2 に変更されます。

ホスト削除前		ホスト削除後	
ホスト ID	ホスト名	ホスト ID	ホスト名
1	SERIALA	1	SERIALA
2	HOST2	2	HOST3
3	HOST3	—	—

WRITEHOSTB の引数 h としてホスト ID を使用している場合は、変更後のホスト ID を指定し直す必要がありますので、注意してください。上記の例の場合、次のようにマクロを変更します。

ホスト削除前のマクロ	WRITEHOSTB(3 ,100,1000,0,\$B10,10);	HOST3 を指定するための 第一引数の「3」を「2」に変更
ホスト削除後のマクロ	WRITEHOSTB(2 ,100,1000,0,\$B10,10);	

ホスト ID は CX-Designer の[通信設定]ダイアログでホスト名の左に表示される番号を確認してください。

- ◆ CX-Designer でホスト名を変更したとき、WRITEHOSTB の引数 h としてホスト名称を使用している場合には、変更後のホスト名に指定し直す必要がありますので、注意してください。

例えば、ホスト名を HOST_1 から HOST_A に変更した場合は、次のように変更します。

ホスト名変更前のマクロ	WRITEHOSTB([HOST_1] ,100,1000,0,\$B10,10);	[HOST_1]を
ホスト名変更後のマクロ	WRITEHOSTB([HOST_A] ,100,1000,0,\$B10,10);	[HOST_A]に変更

NS シリーズ マクロリファレンス

マクロ名	WRITEHOSTW
書式	WRITEHOSTW (h, ch, addr, S, n)
機能	PT メモリ(\$W/\$HW)S から n ワード分のワードデータを、指定したホスト h にコピーします。 h:ホスト(ホスト名/ホスト ID) ch:ホストアドレス種別 *1 addr:ホスト先頭アドレス S:書込元先頭アドレス(\$W0~\$W32767、\$HW0~\$HW8191) n:書込要素数(1~126) *1:章末のアドレス種別番号を参照ください。
リターン値	正常終了:0x0000 異常発生時:上位 8 ビット(B8~B15):MRES(メインレスポンスコード) 下位 8 ビット(B0~B7):SRES(サブレスポンスコード) MRES、SRES については NS シリーズプログラミングマニュアル「5-2-7 通信エラーとその対処」を参照ください。
使用例	ホストに SerialA、SerialB が登録されている場合 ・ホスト ID1 (シリアルポート A) に接続されている PLC の CIO1000 に \$W10~\$W19 のデータを書き込む WRITEHOSTW (1, 100, 1000, \$W10, 10); ・ホスト名称=[SerialB] (シリアルポート B) に接続されている PLC の DM1000 に \$HW10~\$HW19 のデータを書き込む WRITEHOSTW ([SerialB], 300, 1000, \$HW10, 10);

MEMO

- ◆ CX-Designer でホストを削除した場合、削除したホスト以降のホスト ID が前詰めされます。
下表の例では、HOST2 を削除すると、ホスト ID が前詰めされ、HOST3 のホスト ID が 2 に変更されます。

ホスト削除前		ホスト削除後	
ホスト ID	ホスト名	ホスト ID	ホスト名
1	SERIALA	1	SERIALA
2	HOST2	2	HOST3
3	HOST3	—	—

WRITEHOSTW の引数 h としてホスト ID を使用している場合は、変更後のホスト ID を指定し直す必要がありますので、注意してください。上記の例の場合、次のようにマクロを変更します。

ホスト削除前のマクロ	WRITEHOSTW(3 ,100,1000,\$W10,10);	HOST3 を指定するための 第一引数の「3」を「2」に変更
ホスト削除後のマクロ	WRITEHOSTW(2 ,100,1000,\$W10,10);	

ホスト ID は CX-Designer の[通信設定]ダイアログでホスト名の左に表示される番号を確認してください。

- ◆ CX-Designer でホスト名を変更したとき、WRITEHOSTW の引数 h としてホスト名称を使用している場合には、変更後のホスト名に指定し直す必要がありますので、注意してください。

例えば、ホスト名を HOST_1 から HOST_A に変更した場合は、次のように変更します。

ホスト名変更前のマクロ	WRITEHOSTW([HOST_1] ,100,1000,\$W10,10);	[HOST_1]を
ホスト名変更後のマクロ	WRITEHOSTW([HOST_A] ,100,1000,\$W10,10);	[HOST_A]に変更

NS シリーズ マクロリファレンス

マクロ名	BITSET
書式	BITSET (D, c, n)
機能	PT メモリ(\$B/\$HB)のビットアドレス D から n ビット分に c(0/1)を書込みます。 D: 先頭アドレス c: 一括セット値 (0/1) n: 書込要素数 1～32768(\$B) 1～8192(\$HB)
リターン値	なし
使用例	• \$B100 から 10 ビット分(\$B100～\$B109)に値 1 を書き込む BITSET (\$B100, 1, 10); • \$B100 から 10 ビット分(\$B100～\$B109)に値 1 を書き込む \$HB100=1; \$W200=10; BITSET (\$B100, \$HB100, \$W200);

マクロ名	MEMSET
書式	MEMSET (D, c, n)
機能	PT メモリ(\$W/\$HW)のワードアドレス D から n ワード分にデータ c を書込みます。 D: 先頭アドレス c: 一括セット値 -32767～32768(10 進) H0000～HFFFF(16 進) n: 書込要素数 1～32768(\$W) 1～8192(\$HW)
リターン値	なし
使用例	• \$W100 から 10 ワード分(\$W100～\$W109)に値 5 を書き込む MEMSET (\$W100, 5, 10); • \$W100 から 10 ワード分(\$W100～\$W109)に値 5 を書き込む \$HW100=5; \$W200=10; MEMSET (\$W100, \$HW100, \$W200);

NS シリーズ マクロリファレンス

NS シリーズ Ver.8.2 より追加された、マクロの関数詳細は以下のとおりです。

マクロ名	ACOS
書式	ACOS(S)
機能	S の逆余弦値を返します。S は、FLOAT 値として入力します。
リターン値	逆余弦した結果を FLOAT 値(単位:ラジアン)として返します。
使用例	\$W0 = ACOS(\$W10); ‘\$W10(FLOAT 値(2ワード))の ACOS 結果を FLOAT 値として\$W0(2ワード)にセット \$W100= ACOS(-1.0); ‘-1.0 の ACOS 結果を FLOAT 値として\$W100(2ワード)にセット

マクロ名	ASIN
書式	ASIN(S)
機能	S の逆正弦値を返します。S は、FLOAT 値として入力します。
リターン値	逆正弦した結果を FLOAT 値(単位:ラジアン)として返します。
使用例	\$W0 = ASIN(\$W10); ‘\$W10(FLOAT 値(2ワード))の ASIN 結果を FLOAT 値として\$W0(2ワード)にセット \$W100= ASIN(0.5); ‘0.5 の ASIN 結果を FLOAT 値として\$W100(2ワード)にセット

マクロ名	ATAN
書式	ATAN(S)
機能	S の逆正接値を返します。S は、FLOAT 値として入力します。
リターン値	逆正接した結果を FLOAT 値(単位:ラジアン)として返します。
使用例	\$W0 = ATAN(\$W10); ‘\$W10(FLOAT 値(2ワード))の ATAN 結果を FLOAT 値として\$W0(2ワード)にセット \$W100= ATAN(1.233669); ‘1.233669 の ATAN 結果を FLOAT 値として\$W100(2ワード)にセット

マクロ名	ATAN2
書式	ATAN2(S1,S2)
機能	S1,S2 の 2 変数の逆正接値を返します。S1,S2 は、FLOAT 値として入力します。 S1:Y 座標 S2:X 座標
リターン値	2 変数の逆正接した結果を FLOAT 値(単位:ラジアン)として返します。
使用例	\$W0 = ATAN2(\$W10,\$W20); ‘\$W10(FLOAT 値(2ワード))と\$W20(FLOAT 値(2ワード))の ATAN2結果を FLOAT 値として\$W0(2ワード)にセット \$W100= ATAN2(1.233669, 9.258369); ‘1.233669、9.258369 の ATAN2結果を FLOAT 値として\$W100(2ワード)にセット

NS シリーズ マクロリファレンス

マクロ名	CEIL
書式	CEIL(S)
機能	変数 S を切り上げた値を返します。S は、FLOAT 値として入力します。
リターン値	切り上げた結果を FLOAT 値として返します。
使用例	\$W0 = CEIL(\$W10); ‘\$W10(FLOAT 値(2ワード))の CEIL 結果を FLOAT 値として\$W0(2ワード)にセット \$W100= CEIL(1.233669); ‘1.233669 の CEIL 結果を FLOAT 値として\$W100(2ワード)にセット

マクロ名	COS
書式	COS(S)
機能	S の余弦結果を返します。S は、FLOAT 値として入力します。
リターン値	余弦した結果を FLOAT 値(単位:ラジアン)として返します。
使用例	\$W0 = COS(\$W10); ‘\$W10(FLOAT 値(2ワード))の COS 結果を FLOAT 値として\$W0(2ワード)にセット \$W100= COS(0.0); ‘0.0 の COS 結果を FLOAT 値として\$W100(2ワード)にセット

マクロ名	DEG2RAD
書式	DEG2RAD(S)
機能	変数 S を度単位から等価なラジアン単位に変換した値を返します。 S は、FLOAT 値として入力します。
リターン値	結果を FLOAT 値として返します。
使用例	\$W0 = DEG2RAD(\$W10); ‘\$W10(FLOAT 値(2ワード))の値を DEG2RAD した結果を FLOAT 値として\$W0(2ワード)にセット \$W100= DEG2RAD(1.233669); ‘1.233669 の DEG2RAD 結果を FLOAT 値として\$W100(2ワード)にセット

マクロ名	EXP
書式	EXP(S)
機能	S の累乗した結果を返します。S は、FLOAT 値として入力します。
リターン値	結果を FLOAT 値として返します。
使用例	\$W0 = EXP(\$W10); ‘\$W10(FLOAT 値(2ワード))の EXP した結果を FLOAT 値として\$W0(2ワード)にセット \$W100= EXP(-0.369147); ‘-0.369147 の EXP 結果を FLOAT 値として\$W100(2ワード)にセット

NS シリーズ マクロリファレンス

マクロ名	FLOOR
書式	FLOOR(S)
機能	変数 S を切り捨てた値を返します。S は、FLOAT 値として入力します。
リターン値	結果を FLOAT 値として返します。
使用例	\$W0 = FLOOR(\$W10); ‘\$W10 (FLOAT 値 (2ワード)) の値を FLOOR した結果を FLOAT 値として \$W0 (2ワード) にセット \$W100= FLOOR(-0.4554356); ‘-0.4554356 の FLOOR 結果を FLOAT 値として \$W100 (2ワード) にセット

マクロ名	LOG
書式	LOG(S)
機能	S の自然対数結果を返します。S は、FLOAT 値として入力します。
リターン値	結果を FLOAT 値として返します。
使用例	\$W0 = LOG(\$W10); ‘\$W10 (FLOAT 値 (2ワード)) の値を LOG した結果を FLOAT 値として \$W0 (2ワード) に セット \$W100= LOG(6.852943); ‘6.852943 の LOG 結果を FLOAT 値として \$W100 (2ワード) にセット

マクロ名	LOG10
書式	LOG10(S)
機能	S の常用対数 (10 を底とする対数) log10 値を返します。S は、FLOAT 値として入力します。
リターン値	結果を FLOAT 値として返します。
使用例	\$W0 = LOG10(\$W10); ‘\$W10 (FLOAT 値 (2ワード)) の値を LOG10 した結果を FLOAT 値として \$W0 (2ワード) にセット \$W100= LOG10(6.987123); ‘6.987123 の LOG10 した結果を FLOAT 値として \$W100 (2ワード) にセット

マクロ名	RAD2DEG
書式	RAD2DEG(S)
機能	S を度単位にした結果を返します。S は、FLOAT 値として入力します。
リターン値	結果を FLOAT 値として返します。
使用例	\$W0 = RAD2DEG(\$W10); ‘\$W10 (FLOAT 値 (2ワード)) の値を RAD2DEG した結果を \$W0 (2ワード) にセット \$W100= RAD2DEG(-0.236987); ‘-0.236987 の RAD2DEG 結果を FLOAT 値として \$W100 (2ワード) にセット

NS シリーズ マクロリファレンス

マクロ名	SQRT
書式	SQRT(S)
機能	S を平方根した値を返します。S は、FLOAT 値として入力します。
リターン値	結果を FLOAT 値として返します。
使用例	\$W0 = SQRT(\$W10); ‘\$W10(FLOAT 値(2ワード))の値を SQRT した結果を FLOAT 値として\$W0(2ワード)にセット \$W100= SQRT(1.040566); ‘1.040566 の SQRT した結果を FLOAT 値として\$W100(2ワード)にセット

マクロ名	SIN
書式	SIN(S)
機能	S を正弦した結果を返します。S は、FLOAT 値として入力します。
リターン値	結果を FLOAT 値(単位:ラジアン)として返します。
使用例	\$W0 = SIN(\$W10); ‘\$W10(FLOAT 値(2ワード))の値を SIN した結果を FLOAT 値として\$W0(2ワード)にセット \$W100= SIN(-8.660942); ‘-8.660942 を SIN 結果を FLOAT 値として\$W100(2ワード)にセット

マクロ名	TAN
書式	TAN(S)
機能	S を正接した値を返します。S は、FLOAT 値として入力します。
リターン値	結果を FLOAT 値(単位:ラジアン)として返します。
使用例	\$W0 = TAN(\$W10); ‘\$W10(FLOAT 値(2ワード))の値を TAN した結果を FLOAT 値として\$W0(2ワード)にセット \$W100= TAN(-5.731245); ‘-5.731245 の TAN した結果を FLOAT 値として\$W100(2ワード)にセット

マクロ名	POW(S1,S2)
書式	POW(S1,S2)
機能	S1を S2 で累乗した結果を返します。S1,S2 は、FLOAT 値として入力します。
リターン値	結果を FLOAT 値として返します。
使用例	\$W0 = POW(\$W10,\$W20); ‘\$W10,\$W20(FLOAT 値(2ワード))の値を POW 関数で計算した結果を\$W0(2ワード)にセット \$W100= POW(4.238976, 0.001411); ‘4.238976 を 0.001411 で累乗した結果を FLOAT 値として\$W100(2ワード)にセット

NS シリーズ マクロリファレンス

マクロ名	FADD
書式	FADD(S1,S2)
機能	S1 と S2 の和を返します。S1,S2 は、FLOAT 値として入力します。
リターン値	結果を FLOAT 値として返します。
使用例	\$W0 = FADD(\$W10,\$W20); ‘\$W10,\$W20 (FLOAT 値 (2ワード)) の値を足した結果を FLOAT 値として \$W0 (2ワード) にセット \$W100= FADD(2.524563, -1.255669); ‘2.524563 と-1.255669 を足した結果を FLOAT 値として\$W100 (2ワード) にセット

マクロ名	FCOMP
書式	FCOMP(S1,S2)
機能	S1 と S2 を比較した値を返します。S1,S2 は、FLOAT 値として入力します。
リターン値	結果を WORD 値として返します。(0: S1 と S2 が同じ / -1: S1<S2 / 1: S1>S2)
使用例	\$W0 = FCOMP(\$W10,\$W20); ‘\$W10,\$W20 (FLOAT 値 (2ワード)) の値を比較した結果を WORD 値として \$W0 (1ワード) にセット \$W100= FCOMP(2.524563, -1.255669); ‘2.524563 と-1.255669 を比較した結果を WORD 値として\$W100 (1ワード) にセット

マクロ名	FDIV
書式	FDIV(S1,S2)
機能	S1 と S2 の商を返します。S1,S2 は、FLOAT 値として入力します。
リターン値	結果を FLOAT 値として返します。
使用例	\$W0 = FDIV(\$W10,\$W20); ‘\$W10,\$W20 (FLOAT 値 (2ワード)) の値を除算した結果を FLOAT 値として \$W0 (2ワード) にセット \$W100= FDIV(2.524563, -1.255669); ‘2.524563 と-1.255669 を除算した結果を FLOAT 値として\$W100 (2ワード) にセット

マクロ名	FMUL
書式	FMUL(S1,S2)
機能	S1 と S2 の積を返します。S1,S2 は、FLOAT 値として入力します。
リターン値	結果を FLOAT 値として返します。
使用例	\$W0 = FMUL(\$W10,\$W20); ‘\$W10,\$W20 (FLOAT 値 (2ワード)) の値を乗算した結果を FLOAT 値として \$W0 (2ワード) にセット \$W100= FMUL(2.524563, -1.255669); ‘2.524563 と-1.255669 を乗算した結果を FLOAT 値として\$W100 (2ワード) にセット

NS シリーズ マクロリファレンス

マクロ名	FSUB
書式	FSUB(S1,S2)
機能	S1 と S2 の差を返します。S1,S2 は、FLOAT 値として入力します。
リターン値	結果を FLOAT 値として返します。
使用例	\$W0 = FSUB(\$W10,\$W20); ‘\$W10,\$W20 (FLOAT 値 (2ワード)) の値を引き算した結果を FLOAT 値として \$W0 (2ワード) にセット \$W100= FSUB(2.524563, -1.255669); ‘2.524563 と-1.255669 を引き算した結果を FLOAT 値として\$W100 (2ワード) にセット

マクロ名	FSET
書式	FSET(S)
機能	S の値を返します。
リターン値	結果を FLOAT 値として返します。
使用例	\$W0 = FSET(\$W10); ‘\$W10 (FLOAT 値 (2ワード)) の値を FLOAT 値として\$W0 (2ワード) にセット \$W100= FSET(2.524563); ‘2.524563 を FLOAT 値として\$W100 (2ワード) にセット

NS シリーズ マクロリファレンス

NS シリーズ Ver.8.5 より追加された、マクロの関数詳細は以下のとおりです。

マクロ名	LOCALTIME
書式	LOCALTIME(S,D)
機能	S に格納されているローカル時間(ns 単位)をカレンダー時間(年月日時分秒)に変換し、D に出力します。出力されるカレンダー時間は GMT となります。 ※S に格納するローカル時間は 2038 年 01 月 19 日 03:14:07 を越えることはできません。
リターン値	0:正常終了 -1:S または D に範囲外の値を設定した -2:ローカル時間の上限を超えた値を設定した
使用例	READCMEM(\$W0,[DATEANDTIME],4); \$W100=LOCALTIME(\$W0,\$W10); SETTIME(\$W10); ‘変数 DATEANDTIME の値をカレンダー時間に変換して\$W10 にセット

NS シリーズ マクロリファレンス

アドレス種別番号

アドレス種別名称	アドレス種別番号	
	BCD	バイナリ
PT メモリー-\$B	0	0
PT メモリー-\$W	1	1
PT メモリー-\$SB	2	2
PT メモリー-\$SW	3	3
PT メモリー-\$HB	4	4
PT メモリー-\$HW	5	5
リレー (CIO)	100	64
保持リレー (HR)	101	65
補助記憶リレー (AR)	102	66
リンクリレー (LR)	103	67
内部補助リレー (WR)	104	68
タイマ (TIM) (*1)	200	C8
カウンタ (CNT) (*1)	201	C9
データメモリ (DM)	300	12C
拡張データメモリ (EM)	301	12D
拡張データメモリ 0 (EM0)	302	12E
拡張データメモリ 1 (EM1)	303	12F
拡張データメモリ 2 (EM2)	304	130
拡張データメモリ 3 (EM3)	305	131
拡張データメモリ 4 (EM4)	306	132
拡張データメモリ 5 (EM5)	307	133
拡張データメモリ 6 (EM6)	308	134
拡張データメモリ 7 (EM7)	309	135
拡張データメモリ 8 (EM8)	310	136
拡張データメモリ 9 (EM9)	311	137
拡張データメモリ A (EMA)	312	138
拡張データメモリ B (EMB)	313	139
拡張データメモリ C (EMC)	314	13A
拡張データメモリ D (EMD)	315	13B
拡張データメモリ E (EME)	316	13C
拡張データメモリ F (EMF)	317	13D
拡張データメモリ 10 (EM10)	318	13E
拡張データメモリ 11 (EM11)	319	13F
拡張データメモリ 12 (EM12)	320	140
拡張データメモリ 13 (EM13)	321	141
拡張データメモリ 14 (EM14)	322	142
拡張データメモリ 15 (EM15)	323	143
拡張データメモリ 16 (EM16)	324	144
拡張データメモリ 17 (EM17)	325	145
拡張データメモリ 18 (EM18)	326	146

*1: READHOSTW/WRITEHOSTW のみ指定可

第 3 章 エラーメッセージ一覧

本章では、マクロ機能をプロジェクト、画面および機能部品へ追加するときに実行されるエラーチェックで表示されるエラーメッセージについて説明します。

3-1 エラーメッセージ一覧	3 - 1
----------------------	-------

3-1 エラーメッセージ一覧

エラーチェック実行後、エラー一覧ボックスに表示されるエラーメッセージは以下のとおりです。

エラーメッセージ	内容詳細	例
フォーマットエラー	変数名、関数名及びプログラミング用語以外の記述があります。使用している関数の使い方が正しいかどうか確認してください。	\$W0=ABC+100;
変数名エラー	変数名に誤りがあります。	\$B0:3=1;
(無し)	関数や IF 文などの (がぬけています。	IF \$W0==1)
() の数が不一致	() の数があいません。	IF(\$W0=1)!(\$W1=0
, 位置が不正	不正な位置に , が記述されています。	IF(\$W0==1),(\$W1==0)
関数引数エラー	ビットメモリしか設定できない箇所にワードメモリを設定しているなど、関数の引数に誤りがあります。「第 2 章 関数の説明」の「関数引数対応表」で設定可能な引数をご確認ください。	\$W0=BCD(\$B0);
= 命令エラー	3=10 や \$ B0=3 など、代入文に誤りがあります。	\$W0="ABCDE"
プログラム末尾が不正	プログラムが最後まで記述されていません。	\$W10=10+;
IF 文が不整合	IF、ELSE、または ENDIF の記述に誤りがあります。	IF(\$W0==1)!(\$W1==0) \$W10=1; ELSE \$W10=10;
, ; 区切りなし	関数の引数を区切る(,)が不足しています。または、プログラムが(;)で区切られていません。	\$W10=1
FOR 文が不整合	NEXT、BREAK または CONTINUE と対になっていません。	FOR(3) \$W0 = \$W0 + 1;
FOR 文の入れ子オーバー (最大 1)	FOR の入れ子が1つ以上に設定されています。	FOR(3) \$W0 = \$W0 + 1; FOR(5) \$W10 = \$W10 + 10; NEXT; NEXT;

- 本誌に記載のない条件や環境での使用、および原子力制御・鉄道・航空・車両・燃焼装置・医療機器・娯楽機械・安全機器、その他人命や財産に大きな影響が予測されるなど、特に安全性が要求される用途に使用される際には、当社の意図した特別な商品用途の場合や特別の合意がある場合を除き、当社は当社商品に対して一切保証をいたしません。
- 本製品の内、外国為替及び外国貿易法に定める輸出許可、承認対象貨物（又は技術）に該当するものを輸出（又は非居住者に提供）する場合は同法に基づく輸出許可、承認（又は役務取引許可）が必要です。

オムロン株式会社 インダストリアルオートメーションビジネスカンパニー

- 製品に関するお問い合わせ先
お客様相談室

 **0120-919-066**

携帯電話・PHS・IP電話などではご利用いただけませんので、下記の電話番号へおかけください。

電話 **055-982-5015**（通話料がかかります）

■営業時間：8:00～21:00 ■営業日：365日

- FAXやWebページでもお問い合わせいただけます。

FAX **055-982-5051** / www.fa.omron.co.jp

- その他のお問い合わせ

納期・価格・サンプル・仕様書は貴社のお取引先、または貴社担当オムロン販売員にご相談ください。
オムロン制御機器販売店やオムロン販売拠点は、Webページでご案内しています。

オムロン制御機器の最新情報をご覧ください。

www.fa.omron.co.jp

緊急時のご購入にもご利用ください。

オムロン商品のご用命は