OMRON

画像センサ

FH シリーズ

画像処理システム

マクロカスタマイズ機能プログラミングマニュアル

形 FH-2□□□/形 FH-2□□□-□□
形 FH-5□□□/形 FH-5□□□-□□
形 FH-L□□□/形 FH-L□□□-□□



SYSMAC
always in control

SDNB-721S

おことわり

- 本マニュアルの内容の一部または全部を無断で複写、複製、転載することを禁じます。
- 本マニュアルの内容に関しては、改良のため予告なしに仕様などを変更することがあります。あらかじめご了承ください。
- 本マニュアルの内容に関しては、万全を期しておりますが、万一ご不審な点や誤りなど、お気づきのことがありましたら、お手数ですが、弊社支店または営業所までご連絡ください。
  その際、巻末記載の Man.No.（マニュアル No.）も併せてお知らせください。

商標

- Sysmac は、オムロン株式会社 FA 機器製品の日本およびその他の国における商標または登録商標です。
- 本ソフトウェアは、Independent JPEG Group の Code を使用しています。
- 本ソフトウェアには、7-Zip が含まれています。
  7-Zip は、GNU Lesser General Public License の条件にもとづいてライセンスされています。
  GNU Lesser General Public License の詳細については、以下のウェブサイトをご覧ください。
  （www.gnu.org）
- Microsoft、Windows、Windows Vista、Excel、Visual Basic は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。
- Intel、Core は、アメリカ合衆国および / またはその他の国における Intel Corporation の商標です。
- EtherCAT®は、Beckhoff Automation GmbH（ドイツ）よりライセンスを受けた特許取得済み技術であり登録商標です。
- ODVA、CIP、CompoNet、DeviceNet、EtherNet/IP は ODVA の商標です。
- SD ロゴ、SDHC ロゴ、microSD ロゴ、microSDHC ロゴは、SD-3C, LLC の商標です。



- QR コードは、㈱デンソーウェーブの登録商標です。
- MELSEC は、三菱電機株式会社の登録商標です。

その他、記載されている会社名と製品名などにつきましては、各社の登録商標または商標です。

著作権

スクリーンショットは、マイクロソフトの許諾を得て使用しています。

はじめに

このたびは、FH シリーズをお買い上げいただきまして、まことにありがとうございます。
本マニュアルは、FH シリーズを使用する上で、必要な情報を記載しています。
お使いになる前に本マニュアルをよく読んで、FH シリーズの機能・性能などを十分ご理解いただき、システムの構築にご活用ください。
また、お読みになった後も本マニュアルは大切に保管して、いつも手元においてお使いください。

対象となる読者の方々

本マニュアルは、次の方を対象に記述しています。電気の知識（電気工事士あるいは同等の知識）を有する方で

- FA 機器の導入を担当される方
- FA システムを設計される方
- FA 機器を設置、接続される方
- FA 現場を管理される方

対象となる商品

本マニュアルは以下の製品を対象にしています。

- 形 FH-2□□□
- 形 FH-2□□□-□□
- 形 FH-5□□□
- 形 FH-5□□□-□□
- 形 FH-L□□□
- 形 FH-L□□□-□□

FH シリーズ各製品の仕様の一部または制約事項が、他のマニュアルに記載されている場合があります。「分冊構成 (P.2)」および「関連マニュアル (P.17)」をご確認ください。

分冊構成

FH シリーズのマニュアルは、次表のとおりに分冊されています。目的に応じてお読みいただき、FH シリーズを活用してください。

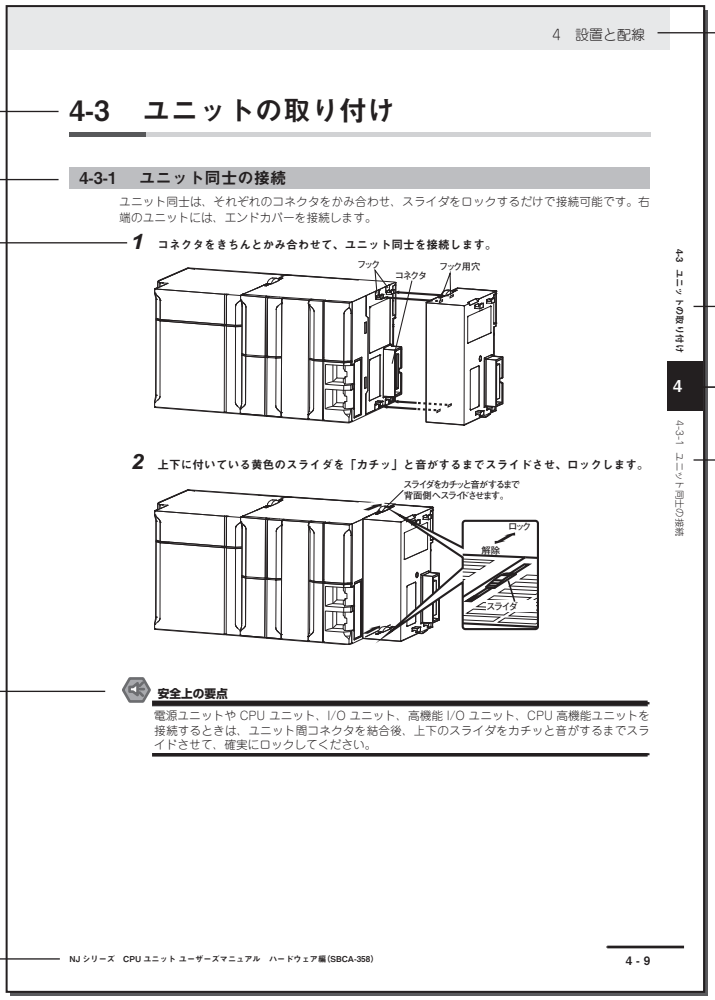
使用目的	マニュアル					
	基本情報					
	画像センサ FH/FHV シリーズ 画像処理システム ユーザーズマニュアル	画像センサ FH シリーズ 画像処理システム ハードウェアセットアップマニュアル	画像センサ FH/FHV シリーズ 画像処理システム 処理項目機能リファレンスマニュアル	画像センサ FH シリーズ 画像処理システム マクロカスタマイズ機能 プログラミングマニュアル	画像センサ FH/FHV シリーズ 画像処理システム ユーザーズマニュアル 通信設定編	画像センサ FH/FHV シリーズ 画像処理システム オペレーションマニュアル Sysmac Studio 編
FH シリーズの概要を知る	●	●				
設置、配線を行う						
EtherCAT を使用する場合						
EtherNet/IP を使用する場合						
PROFINET を使用する場合		●				
Ethernet を使用する場合						
RS-232C を使用する場合						
パラレルインタフェースを使用する場合						
センサコントローラの通信設定を行う						●
EtherCAT を使用する場合						
EtherNet/IP を使用する場合						
PROFINET を使用する場合	●	●			●	
Ethernet を使用する場合						
RS-232C を使用する場合						
パラレルインタフェースを使用する場合						
センサコントローラの設定を行う						●
EtherCAT を使用する場合						
EtherNet/IP を使用する場合						
PROFINET を使用する場合	●				●	
Ethernet を使用する場合						
RS-232C を使用する場合						
パラレルインタフェースを使用する場合						
シーン（計測フロー）を作成、設定する						●
EtherCAT を使用する場合						
EtherNet/IP を使用する場合						
PROFINET を使用する場合	●		●			
Ethernet を使用する場合						
RS-232C を使用する場合						
パラレルインタフェースを使用する場合						

使用目的	マニュアル						
	基本情報						
	画像センサ FH/FHV シリーズ 画像処理システム ユーザーズマニュアル	画像センサ FH シリーズ 画像処理システム ハードウェアセットアップマニュアル	画像センサ FH/FHV シリーズ 画像処理システム 処理項目機能リファレンスマニュアル	画像センサ FH シリーズ 画像処理システム マクロカスタマイズ機能 プログラミングマニュアル	画像センサ FH/FHV シリーズ 画像処理システム ユーザーズマニュアル 通信設定編	画像センサ FH/FHV シリーズ 画像処理システム オペレーションマニュアル Sysmac Studio 編
シーン（計測フロー）を最適化する			●	●		
EtherCAT を使用する場合						
EtherNet/IP を使用する場合						
PROFINET を使用する場合						
Ethernet を使用する場合						
RS-232C を使用する場合						
パラレルインタフェースを使用する場合						
コントローラと接続する	●	●			●	●
EtherCAT を使用する場合						
EtherNet/IP を使用する場合						
PROFINET を使用する場合						
Ethernet を使用する場合						
RS-232C を使用する場合						
パラレルインタフェースを使用する場合						
運用に役立つ機能を使用する	●					●
EtherCAT を使用する場合						
EtherNet/IP を使用する場合						
PROFINET を使用する場合						
Ethernet を使用する場合						
RS-232C を使用する場合						
パラレルインタフェースを使用する場合						
トラブル発生時やエラーメッセージの内容や対処方法を知る	●					

マニュアルの読み方

ページ構成

このマニュアルの各ページの構成は次のとおりです。



節タイトル

4-3　ユニットの取り付け

項タイトル

4-3-1　ユニット同士の接続

操作手順

操作手順を示します。

注記、補足、参照先

注記、補足、参照先などの各項目を、アイコンによって示しています。

マニュアル名称

章タイトル

節タイトル

項タイトル

現在のページの章／節／項タイトルをそれぞれ示します。

章番号

現在のページの章番号を示します。

（注）このページは、説明用の見本です。実際の内容とは異なります。

アイコン

このマニュアルで使われているアイコンには、次のような意味があります。



安全上の要点

製品を安全に使用するために実施または、回避すべきことを示します。



使用上の注意

製品が動作不能、誤動作、または性能や機能への悪影響を予防するために実施または回避すべきことを示します。



参考

必要に応じて読んでいただきたい項目です。
知っておくと便利な情報や、使用する上で参考となる内容について説明しています。

マニュアル内の表記について

本文中では、メニューなどを次のように記載しています。

[]	メニュー	メニューバーなどに表示されているメニュー名や、処理項目名を表します。
「」	項目名	画面に表示されている項目名などを表します。

目次構成

1 マクロカスタマイズ機能概要
2 マクロカスタマイズ機能使用方法
3 画面の見方と設定方法
4 プログラムの基本
5 マクロプログラム
6 デバッグ機能
7 トラブルシューティング
8 マクロ関数
9 マクロリファレンス
I 索引

目次

はじめに **1**
対象となる読者の方々 1
対象となる商品 1

分冊構成 **2**

マニュアルの読み方 **4**
ページ構成 4
アイコン 5
マニュアル内の表記について 5

目次構成 **7**

ご承諾事項 **11**

安全上の注意 **13**

安全上の要点 **14**

使用上の注意 **15**

法規と規格 **16**

関連マニュアル **17**

用語説明 **18**

マニュアル改訂履歴 **22**

第 1 章　マクロカスタマイズ機能概要

1-1　マクロカスタマイズ機能について **1-2**
1-1-1　マクロカスタマイズ機能一覧 1-2
1-1-2　本書の構成 1-4

第 2 章　マクロカスタマイズ機能使用方法

2-1　マクロカスタマイズ機能の使い方について **2-2**
2-1-1　マクロカスタマイズ機能の構成 2-2
2-1-2　マクロカスタマイズ機能の使用手順 2-3

第 3 章　画面の見方と設定方法

3-1　画面の見かたと設定方法について **3-2**
3-1-1　プログラム編集画面の見かた 3-2
3-1-2　システム状態監視コンソールウィンドウの見かた 3-6
3-1-3　ユニット演算マクロ処理項目の設定画面の見かたと設定方法 3-7
3-1-4　シーン制御マクロツールの設定画面の見かたと設定方法 3-12
3-1-5　通信コマンドマクロツールの設定画面の見かたと設定方法 3-13
3-1-6　ユニットマクロ処理項目の設定画面の見かたと設定方法 3-20

3-1-7　プログラムのセーブとロード 3-24

第 4 章　プログラムの基本

4-1　プログラムの基本について 4-2
4-1-1　基本構文 4-2
4-1-2　定数 4-5
4-1-3　変数 4-6
4-1-4　シーン変数、システム変数 4-11
4-1-5　演算子 4-12
4-1-6　式 4-14

第 5 章　マクロプログラム

5-1　マクロプログラムの書き方について 5-2
5-1-1　処理ユニットに関係するデータの種類 5-2
5-1-2　システムに関係するデータの種類 5-7
5-1-3　データのスコープと保存領域 5-9
5-1-4　状態遷移と実行タイミング 5-12
5-1-5　処理の排他制御 5-18

第 6 章　デバッグ機能

6-1　デバッグ機能の使い方について 6-2
6-1-1　デバッグの準備 6-2
6-1-2　デバッグの手順 6-4
6-1-3　エラー発生状況の確認 6-4
6-1-4　デバッグの開始 6-5
6-1-5　エラー原因の特定 6-6
6-1-6　エラーの解消 6-9
6-1-7　デバッグの終了 6-9

第 7 章　トラブルシューティング

7-1　トラブルシューティングについて 7-2
7-1-1　プログラミングするときのトラブルシューティング 7-2
7-1-2　動作確認するときのトラブルシューティング 7-3
7-1-3　デバッグするときのトラブルシューティング 7-7
7-1-4　本運用するときのトラブルシューティング 7-8

第 8 章　マクロ関数

8-1　マクロ関数一覧 8-2
8-1-1　アルファベット順 8-2
8-1-2　インデックス（機能別） 8-13
8-2　コマンド詳細 8-21

第 9 章　マクロリファレンス一覧

9-1　エラー一覧 9-2
9-2　予約一覧 9-5
9-2-1　予約語一覧 9-5
9-2-2　グローバルデータの予約語一覧 9-9

9-3　システムデータ一覧 9-10

9-4　IO モジュール一覧 9-25
9-4-1　EtherCAT 9-26
9-4-2　EtherNetIP 9-28
9-4-3　PROFINET 9-31
9-4-4　ParallelIo 9-34
9-4-5　SerialNormal 9-36
9-4-6　SerialPlcLinkM 9-38
9-4-7　SerialPlcLink 9-40
9-4-8　TcpClient 9-43
9-4-9　TcpNormal 9-45
9-4-10　UdpNormal 9-47
9-4-11　UdpPlcLinkM 9-49
9-4-12　UdpPlcLinkY 9-52
9-4-13　UdpPlcLink 9-55
9-4-14　TcpPlcLinkM 9-58
9-4-15　TcpPlcLink 9-61

9-5　処理項目識別名一覧 9-64
9-5-1　検査・計測をする 9-64
9-5-2　画像を取り込む 9-64
9-5-3　画像を補正する 9-65
9-5-4　検査・計測を補助する 9-65
9-5-5　処理を分岐する 9-66
9-5-6　結果を外部へ出力する 9-66
9-5-7　結果を画面へ表示する 9-66

9-6　図形データ一覧 9-67
9-6-1　図形データの構造一覧 9-67
9-6-2　図形データの配列要素一覧 9-68

9-7　図形番号一覧 9-70

9-8　モデル番号一覧 9-72

9-9　画像番号一覧 9-74

9-10　サブ画像番号一覧 9-77

索引

ご承諾事項

「当社商品」について特別の合意がない場合には、お客様のご購入先にかかわらず、本ご承諾事項記載の条件を適用いたします。

● 定義

本ご承諾事項中の用語の定義は次のとおりです。

- 「当社商品」:「当社」の FA システム機器、汎用制御機器、センシング機器、電子・機構部品
- 「カタログ等」:「当社商品」に関する、ベスト制御機器オムロン、電子・機構部品総合カタログ、その他のカタログ、仕様書、取扱説明書、マニュアル等であって電磁的方法で提供されるものも含みます。
- 「利用条件等」:「カタログ等」に記載の、「当社商品」の利用条件、定格、性能、動作環境、取り扱い方法、利用上の注意、禁止事項その他
- 「お客様用途」:「当社商品」のお客様におけるご利用方法であって、お客様が製造する部品、電子基板、機器、設備またはシステム等への「当社商品」の組み込み、または利用を含みます。
- 「適合性等」:「お客様用途」での「当社商品」の(a)適合性、(b)動作、(c)第三者の知的財産の非侵害、(d)法令の遵守および(e)各種規格の遵守

● 記載事項のご注意

「カタログ等」の記載内容については次の点をご理解ください。

- 定格値および性能値は、単独試験における各条件のもとで得られた値であり、各定格値および性能値の複合条件のもとで得られる値を保証するものではありません。
- 参考データはご参考として提供するもので、その範囲で常に正常に動作することを保証するものではありません。
- 利用事例はご参考ですので、「当社」は「適合性等」について保証いたしかねます。
- 「当社」は、改善や当社都合等により、「当社商品」の生産を中止し、または「当社商品」の仕様を変更することがあります。

● ご利用にあたってのご注意

ご採用およびご利用に際しては次の点をご理解ください。

- 定格・性能ほか「利用条件等」を遵守しご利用ください。
- お客様ご自身にて「適合性等」をご確認いただき、「当社商品」のご利用の可否をご判断ください。「当社」は「適合性等」を一切保証いたしかねます。
- 「当社商品」がお客様のシステム全体の中で意図した用途に対して、適切に配電・設置されていることをお客様ご自身で、必ず事前に確認してください。
- 「当社商品」をご使用の際には、(i)定格および性能に対し余裕のある「当社商品」のご利用、冗長設計などの安全設計、(ii)「当社商品」が故障しても、「お客様用途」の危険を最小にする安全設計、(iii)利用者に危険を知らせるための、安全対策のシステム全体としての構築、(iv)「当社商品」および「お客様用途」の定期的な保守、の各事項を実施してください。
- 「当社」は 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、不正アクセスにより、「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、データベースが感染したとしても、そのことにより直接または間接的に生じた損失、損害その他の費用について一切責任を負わないものとします。
  お客様ご自身にて、①アンチウイルス保護、②データ入出力、③紛失データの復元、④「当社商品」またはインストールされたソフトウェアに対するコンピュータウイルス感染防止、⑤「当社商品」に対する不正アクセス防止についての十分な措置を講じてください。
- 「当社商品」は、一般工業製品向けの汎用品として設計製造されています。従いまして、次に掲げる用途での使用は意図しておらず、お客様が「当社商品」をこれらの用途に使用される際には、「当社」

は「当社商品」に対して一切保証をいたしません。ただし、次に掲げる用途であっても「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は除きます。

a）高い安全性が必要とされる用途（例：原子力制御設備、燃焼設備、航空・宇宙設備、鉄道設備、昇降設備、娯楽設備、医用機器、安全装置、その他生命・身体に危険が及びうる用途）

b）高い信頼性が必要な用途（例：ガス・水道・電気等の供給システム、24 時間連続運転システム、決済システムほか権利・財産を取扱う用途など）

c）厳しい条件または環境での用途（例：屋外に設置する設備、化学的汚染を被る設備、電磁的妨害を被る設備、振動・衝撃を受ける設備など）

d）「カタログ等」に記載のない条件や環境での用途

- 上記の(a)から(d)に記載されている他、「本カタログ等記載の商品」は自動車（二輪車含む。以下同じ）向けではありません。自動車に搭載する用途には利用しないで下さい。自動車搭載用商品については当社営業担当者にご相談ください。

● 保証条件

「当社商品」の保証条件は次のとおりです。

- 保証期間　ご購入後 1 年間といたします。
  （ただし「カタログ等」に別途記載がある場合を除きます。）
- 保証内容　故障した「当社商品」について、以下のいずれかを「当社」の任意の判断で実施します。
  a）当社保守サービス拠点における故障した「当社商品」の無償修理
  （ただし、電子・機構部品については、修理対応は行いません。）
  b）故障した「当社商品」と同数の代替品の無償提供
- 保証対象外　故障の原因が次のいずれかに該当する場合は、保証いたしません。
  a）「当社商品」本来の使い方以外のご利用
  b）「利用条件等」から外れたご利用
  c）本ご承諾事項「ご利用にあたってのご注意」に反するご利用
  d）「当社」以外による改造、修理による場合
  e）「当社」以外の者によるソフトウェアプログラムによる場合
  f）「当社」からの出荷時の科学・技術の水準では予見できなかった原因
  g）上記のほか「当社」または「当社商品」以外の原因（天災等の不可抗力を含む）

● 責任の制限

本ご承諾事項に記載の保証が、「当社商品」に関する保証のすべてです。
「当社商品」に関連して生じた損害について、「当社」および「当社商品」の販売店は責任を負いません。

● 輸出管理

「当社商品」または技術資料を、輸出または非居住者に提供する場合は、安全保障貿易管理に関する日本および関係各国の法令・規制を遵守ください。お客様が法令・規則に違反する場合には、「当社商品」または技術資料をご提供できない場合があります。
（輸出貿易管理令該当品）
形 FH-SMX-SWIR／形 FH-SMX01-SWIR は、外国為替および外国貿易法の規定により、輸出の許可が必要な貨物・技術に該当します。
日本国外に持ち出す際には、日本国政府の輸出許可申請など、必要な手続きをお取りください。
輸出先国から持ち出す際には、当該国政府の輸出貿易管理令等を確認し、必要な手続きをお取りください。

安全上の注意

安全上の注意の内容については、以下のマニュアルを参照してください。詳細は、『画像処理システム FH/FHV シリーズ ユーザーズマニュアル（SDNB-719)』の「安全上の注意」を参照してください。

安全上の要点

安全上の要点の内容については、以下のマニュアルを参照してください。詳細は、『画像処理システム FH/FHV シリーズ ユーザーズマニュアル（SDNB-719)』の「安全上の要点」を参照してください。

使用上の注意

使用上の注意の内容については、以下のマニュアルを参照してください。詳細は、『画像処理システム FH/FHV シリーズ ユーザーズマニュアル（SDNB-719)』の「使用上の注意」を参照してください。

法規と規格

法規と規格の内容については、以下のマニュアルを参照してください。詳細は、『画像処理システム FH/FHV シリーズ ユーザーズマニュアル（SDNB-719)』の「法規と規格」を参照してください。

関連マニュアル

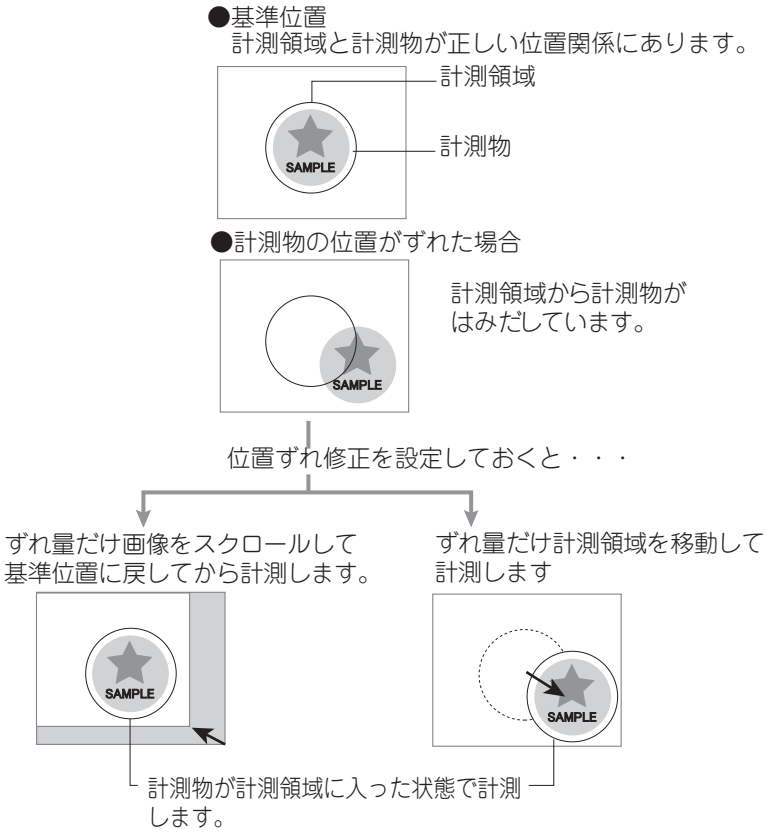
関連するマニュアルは、下表のとおりです。あわせてご覧ください。

マニュアル名称	Cat.No.	形式	用途	内容
画像処理システム FH 取扱説明書	3648743-1	形 FH-2□□□ 形 FH-2□□□-□□ 形 FH-5□□□ 形 FH-5□□□-□□	画像センサ FH シリーズのセンサコントローラの安全上、使用上の注意事項の確認時	FH シリーズの安全上の注意、使用上の注意、ご承諾事項などについて記載しています。
画像処理システム FH-L 取扱説明書	3615792-0	形 FH-L□□□ 形 FH-L□□□-□□	画像センサ FH-L シリーズのセンサコントローラの安全上、使用上の注意事項の確認時	FH-L シリーズの安全上の注意、使用上の注意、ご承諾事項などについて記載しています。
画像処理システム FH/FHV シリーズ ユーザーズマニュアル	SDNB-719	形 FH-2□□□ 形 FH-2□□□-□□ 形 FH-5□□□ 形 FH-5□□□-□□ 形 FH-L□□□ 形 FH-L□□□-□□	画像センサ FH/FHV シリーズのセンサコントローラの設定方法を知りたいとき	画像センサ FH/FHV シリーズのセンサコントローラを使用するためのソフト的な機能、設定、操作について記載しています。
画像処理システム FH/FHV シリーズ 処理項目機能リファレンスマニュアル	SDNB-713		計測フローの設計・操作時に、各処理項目の内容を確認するとき	FH/FHV シリーズで使用できる各処理項目の機能、設定、操作について記載しています。
画像処理システム FH/FHV シリーズ ユーザーズマニュアル 通信設定編	SDNB-714		通信機能の設定や確認するとき	FH/FHV シリーズを PLC などの外部装置と通信するための、機能、設定および通信方法について記載しています。 以下の通信プロトコルを記載しています。 パラレル、PLC リンク、EtherNet/IP、EtherCAT、無手順
画像処理システム FH シリーズ ハードウェアセットアップマニュアル	SDNB-720		画像センサコントローラ FH シリーズのセットアップやハードウェアの仕様を知りたいとき	FH シリーズの仕様、外形寸法、各部の名称、I/O、取付、配線について記載しています。
画像処理システム FH シリーズ マクロカスタマイズ機能プログラミングマニュアル	SDNB-721		マクロカスタマイズ機能を使用した操作やプログラミングするとき	FH シリーズのマクロカスタマイズ機能を使用するための機能、設定、操作について記載しています。
画像処理システム FH/FHV シリーズ オペレーションマニュアル Sysmac Studio 編	SDNB-715	形 FH-2□□□ 形 FH-2□□□-□□ 形 FH-5□□□ 形 FH-5□□□-□□	NJ/NX シリーズと EtherCAT 通信するとき	Sysmac Studio FH/FHV ツールから FH/FHV シリーズの設定／操作をする場合の操作方法について記載しています。

用語説明

用語	内容
FH シリーズ	形 FH-2 □□□／形 FH-2 □□□ - □□／形 FH-5 □□□／形 FH-5 □□□ - □□／形 FH-L □□□／形 FH-L □□□ - □□のシリーズ全体を表す呼称です。
FH-2000 シリーズ	形 FH-2 □□□／形 FH-2 □□□ - □□のシリーズ全体を表す呼称です。
FH-5000 シリーズ	形 FH-5 □□□／形 FH-5 □□□ - □□のシリーズ全体を表す呼称です。
FH-L シリーズ	形 FH-L □□□／形 FH-L □□□ - □□のシリーズ全体を表す呼称です。
FHV シリーズ	形 FHV シリーズ全体を表す呼称です。
FZ5 シリーズ	形 FZ5-6 □□／形 FZ5-6 □□ - □□／形 FZ5-8 □□／形 FZ5-8 □□ - □□／形 FZ5-11 □□／形 FZ5-11 □□ - □□／形 FZ5-12 □□／形 FZ5-12 □□ - □□／形 FZ5-L35 □／形 FZ5-L35 □ - □□のシリーズ全体を表す呼称です。
FZ5-600 シリーズ	形 FZ5-6 □□／形 FZ5-6 □□ - □□のシリーズ全体を表す呼称です。
FZ5-800 シリーズ	形 FZ5-8 □□／形 FZ5-8 □□ - □□のシリーズ全体を表す呼称です。
FZ5-1100 シリーズ	FZ5-1100 シリーズ形 FZ5-11 □□／形 FZ5-11 □□ - □□のシリーズ全体を表す呼称です。
FZ5-1200 シリーズ	形 FZ5-12 □□／形 FZ5-12 □□ - □□のシリーズ全体を表す呼称です。
FZ5-L シリーズ	形 FZ5-L35 □／形 FZ5-L35 □ - □□のシリーズ全体を表す呼称です。
センサコントローラ	FH/FZ5 シリーズの総称です。
計測フロー （または略して「フロー」）	計測処理の一連の流れを指します。処理項目を、組み合わせて作成したシーンから成ります。
計測処理	検査・計測の処理項目を実行すること。
計測 ID	計測時のセンサコントローラの日時情報にライン番号を合わせた値です。 計測 ID のフォーマット：YYYY-MM-DD_HH-MM-SS-XXXN YYYY: 西暦、MM: 月、DD: 日、HH: 時、MM: 分、SS: 秒、XXX: ミリ秒、N: ライン番号になります。 • 例 計測時間 2007 年 12 月 24 日 11 時 10 分 25.500 秒でライン 0 の場合、計測 ID は「2007-12-24_11-10-25-5000」となります。
処理項目	自由な組み合わせが可能なように分割され、かつパッケージ化された、画像処理検査の項目単位です。 例えば、サーチ、位置ずれ修正、ファインマッチングなどがあります。 画像入力、検査・計測、画像補正、検査・計測補助、処理分岐、結果外部出力、結果画面表示などに大分類できます。 これらを自由に組み合わせることで、さまざまなアプリケーションに対応することができます。 この処理項目をユニットに登録することで、シーン（計測フローの段取り替え用の単位）を作成します。
シーン	処理項目を組み合わせて作成した、計測フローの段取り替え用の単位です。 計測する場面（計測物の種類および検査内容）に相当するため、シーンと呼びます。 計測ごと、または計測の内容ごとに、このシーンを作成しておきます。 計測の対象物や検査内容が変わったとき、シーンを切り替えるだけで、段取り変えを行うことができます。 シーンは、128 個まで設定できます。128 個を超えるシーンが必要なときは、複数のシーングループに分けるか、シーングループデータ変換ツールを使用して、129 個以上のシーンから成るシーングループを作成します。
処理ユニット （または略して、「ユニット」）	シーンに登録された処理項目です。 上から順に自動で番号が振られ、その順で実行されます。 この処理ユニットに処理項目を登録して、シーン（計測フローの段取り変え用の単位）を作成します。
計測トリガ	計測を実行させるタイミングです。（STEP 信号など） （計測トリガ以外に、計測コマンドでの計測実行もできます。）

用語	内容
テスト計測	表示中のシーンに設定されている条件で、手動で計測のテスト（確認）を行うことです。 調整画面で実行可能です。スマートカメラ内部で完結され、計測結果は外部インタフェースには出力されません。 ただし、［テスト計測設定］ の「外部出力あり」にチェックをしていると、計測実行後、計測結果を出力します。
単発計測	トリガ入力に同期して、1 回だけ計測を実行します。
連続計測	トリガ入力なしで、自動的に計測を繰り返し実行します。
動作モード	• 倍速マルチインプット 1 回目のトリガに対する計測フロー処理、2 回目のトリガに対する計測フロー処理を並行して実施することで、高速なトリガ入力間隔を実現した高速モードです。マルチインプット機能とセットで使用します。 • 複数ラインランダム 複数の計測フローを独立して処理することができるトリガモードです。 従来の画像処理では、2 つ以上のトリガを同時に受け付けることはできませんでした。複数ラインランダムトリガモードは、1 つのコントローラで複数系統のトリガをランダムに入力し、複数のシーンを並列かつ独立して処理させることが可能です。 • ノンストップ調整 計測を実施しながら、フローや設定パラメータの調整を、実施することができるモードです。 これにより、ラインや検査を停止せずに調整ができます。 • 標準 計測とロギングを、並列処理させることができるロギングモードです。 従来は、計測処理中はロギングができなかったため、計測とロギングのどちらか一方を優先させ、他方は待機をしていました。 計測時間に影響を与えることなく、外部ストレージに計測画像を保存することができます。
並列処理 （上記動作モード共通オプション）	計測フローの一部を 2 以上のタスクに分割して、各タスクを並列に処理することで、計測時間を短縮することができます。 並列処理用の処理項目を使用して、並列処理を任意に指定することができます。
マルチインプット機能	マルチインプット機能連続して高速に画像入力できる機能です。 画像入力が完了した時点で、次の STEP を受け付けることができます。計測処理完了まで待つ必要がありません。 画像入力が完了したかどうかは、READY 信号の状態で確認できます。計測処理を実行中でも、READY が ON であれば、次の STEP を受け付けることができます。

用語	内容
位置ずれ修正	計測物の位置や向きが一定でないときに、基準位置と現在の位置ずれ量を算出し、補正してから計測を行うことです。 位置修正関連の処理項目の中から、計測物に合ったものを選んでください。 
基準位置	常に基準となる点です。モデル登録した位置が基準と異なる場合、基準位置タブにて設定を変更します。
モデル	検査対象とする画像パターンです。対象物の画像から、特徴的な部分を切り取って、モデルとして登録します。

用語	内容
2 の補数	2 進数による負数の表現方法として一般的に使われているものです。 「正数の全ビットを反転させ、それに 1 を加算した値」で負数を表します。 例:「-1」を、2 の補数で表現する 「-1」は、「0-1」で算出できます。 （1があると想定して引き算してみてください） 00000000 (= 0) −) 00000001 (= 1) 11111111 (=-1) ⇦ -1を「2の補数」で表現したもの（8ビットの場合） このような計算をしなくても簡単に算出する方法があります。 それが、「負数= 正数の全ビットを反転させ、それに 1 を加算した値」です。 00000001 (= 1) ↓全ビットを反転させる 11111110 ↓1を加算する 11111111 (=-1) 「最上位桁」で、正数・負数を判断できます。 • 0 の場合：正数（または 0） • 1 の場合：負数 2 の補数には、正負の数をそのまま演算できるという利点があります。 11111111 (= -1) +)00001010 (= 10) 00001001 (= 9)

マニュアル改訂履歴

マニュアル改訂記号は、表紙と裏表紙の下に記載されている Man. No. の末尾に付記されます。



改訂記号	改訂年月	改訂内容	ソフトウェアバージョン
A	2016 年 4 月	初版	Ver.5.6
B	2017 年 3 月	誤記修正	Ver.5.71
C	2017 年 6 月	誤記修正	Ver.5.72
D	2018 年 7 月	ソフトウェアバージョンアップによる説明の追加	Ver.6.1
E	2019 年 7 月	• ソフトウェアバージョンアップによる説明の追加 • FZ5 シリーズを削除	Ver.6.3
F	2019 年 11 月	レイアウト調整 誤記修正	Ver.6.3
G	2020 年 6 月	ソフトウェアバージョンアップによる説明の追加	Ver.6.4
H	2020 年 11 月	誤記修正	Ver.6.4
J	2021 年 6 月	誤記修正	Ver.6.4
K	2022 年 5 月	形 FH-1050、形 FH-1050-10、形 FH-1050-20、形 FH-3050、形 FH-3050-10、形 FH-3050-20 削除 ソフトウェアバージョンアップによる説明の追加 誤記修正	Ver.6.5
L	2022 年 12 月	「関連マニュアル」更新 誤記修正	Ver.6.5
M	2023 年 3 月	誤記修正	Ver.6.5
N	2024 年 3 月	ソフトウェアバージョンアップによる説明の追加（8-2 コマンド詳細、9-1-3 システムデータ一覧、9-1-9 画像番号一覧） 「関連マニュアル」更新 誤記修正	Ver.6.55 Ver.6.6
P	2025 年 6 月	形 FH-2050/FH-2050-10/FH-2050-20 追削除、形 FH-5050/FH-5050-10/FH-5050-20 削除、形 FH-5550/FH-5550-10/FH-5550-20 削除、形 FH-L550/FH-L550-10 削除 ソフトウェアバージョンアップによる説明の追加（3-1-1 プログラム編集画面の見かた、3-1-3 ユニット演算マクロ処理項目の設定画面の見かたと設定方法、3-1-4 シーン制御マクロツールの設定画面の見かたと設定方法、ExecuteImageLogging、Measure、SaveMeasureImage、9-1-2 予約一覧、9-1-3 システムデータ一覧、9-1-5 処理項目識別名一覧、9-1-7 図形番号一覧、9-1-9 画像番号一覧、9-1-10 サブ画像番号一覧） 誤記修正	Ver.6.7

改訂記号	改訂年月	改訂内容	ソフトウェアバージョン
R	2025 年 11 月	レイアウト調整（9 章） 誤記修正	Ver.6.7
S	2026 年 9 月	ソフトウェアバージョンアップによる説明の追加（SystemInfo、9-1-3 システムデーター覧） 誤記修正	Ver.6.8

1

マクロカスタマイズ機能概要

1-1　マクロカスタマイズ機能について 1-2
1-1-1　マクロカスタマイズ機能一覧 1-2
1-1-2　本書の構成 1-4

1-1 マクロカスタマイズ機能について

FH シリーズでは、マクロカスタマイズ機能を活用すると、よりきめ細かく拡張性のある画像処理を実現できます。
マクロカスタマイズ機能とは、通常の「演算」処理項目よりも、高機能な各種の演算、計測フロー／シーンの制御、通信コマンドの作成、さらに各種の表示制御、結果出力制御などができる機能です。
FH シリーズでは、マクロカスタマイズ機能として、次のような機能を使用することができます。

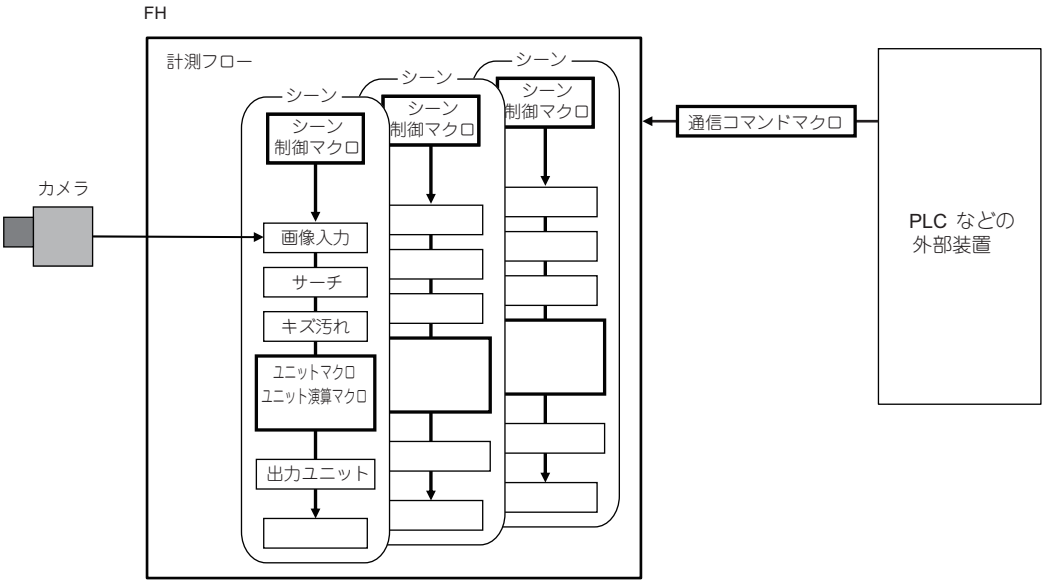
- 「ユニット演算マクロ」処理項目
- 「シーン制御マクロ」ツール
- 「通信コマンドマクロ」ツール
- 「ユニットマクロ」処理項目

1-1-1 マクロカスタマイズ機能一覧

マクロカスタマイズ機能の一覧と、おおよその難易度の目安です。機能により、作成の難易度が異なるので、実現したい拡張と難易度の目安をご確認の上ご利用ください。

難易度の目安	機能	概要	対象機能			
			計測処理		通信コマンド	シーン制御
			演算	演算以外*1		
高 ↑	ユニットマクロ処理項目	処理ユニットの計測処理を補完・拡張できる機能です。 計測処理だけでなく、結果表示処理をはじめ、シーン切替時に処理される計測初期化処理などで、ユーザ独自の処理を作成することができます。	○	○	-	-
	通信コマンドマクロツール	通信コマンドを補完･拡張できる機能です。 ユーザ独自の通信コマンドを作成することができます。 標準の通信コマンドにない機能を作成する、複数の通信コマンド機能をひとつのの通信コマンドにまとめるなどの使い方ができます。	-	-	○	-
	シーン制御マクロツール	計測フローやシーンの制御を補完・拡張できる機能です。 計測フローに処理ユニットを追加して設定するなどの使い方ができます。	-	-	-	○
↓ 低	ユニット演算マクロ処理項目	計測時の演算処理を補完・拡張できる機能です。 「演算」処理項目では、設定が難しい複雑な演算をする、論理式や繰り返し処理を含む演算をするなどの使い方ができます。	○	-	-	-

*1. 結果表示（グラフィック表示、詳細テキスト表示など）、結果出力、初期化処理の作成



次のようなとき、マクロカスタマイズ機能を使用します。

こんなとき	使用するマクロカスタマイズ機能
「演算」処理項目では難しいまたは実現できない、以下のような演算処理をするとき • 複数行にわたる計算式、論理式、条件分岐、ループ処理、データを設定する処理をするとき • 演算結果に対する判定処理を独自に定義するとき	ユニット演算マクロ処理項目
「演算」処理項目や、「ユニット演算マクロ」処理項目では難しいまたは実現できない、以下のような処理のカスタマイズをするとき、またはその処理を含むとき • 結果表示（グラフィック表示、詳細テキスト表示など） • 結果出力 • 計測初期化処理（シーン切替時の処理など）	ユニットマクロ処理項目
以下のようなシーンの制御をするとき • 複数の処理ユニットの設定を一括で変更するとき • 複数の処理ユニットに共通のデータを管理するとき • 処理ユニットを追加／削除するとき	シーン制御マクロツール
標準の通信コマンドでは難しいまたは実現できない、以下のような拡張をするとき • 標準の通信コマンドにはない機能が必要なとき • 複数の通信コマンド機能をひとつの通信コマンドにまとめるとき	通信コマンドマクロツール

1-1-2 本書の構成

本書の各内容とマクロカスタマイズ機能の関係は、以下のとおりです。使用する機能項目で、必要な項目を参照してください。

項目名	マクロカスタマイズ機能			
	ユニット演算マクロ	シーン制御マクロ	通信コマンドマクロ	ユニットマクロ
マクロカスタマイズ機能とは 参照：「第 1 章 マクロカスタマイズ機能概要 (P.1-1)」	必要			
マクロカスタマイズ機能の使い方 参照：「第 2 章 マクロカスタマイズ機能使用方法 (P.2-1)」	必要			
ユニット演算マクロ処理項目を使うための準備 参照：「ユニット演算マクロ処理項目の使用手順 (P.2-4)」	必要	不要		
シーン制御マクロを使うための準備 参照：「シーン制御マクロツールの使用手順 (P.2-4)」	不要	必要	不要	
通信コマンドマクロツールを使うための準備 参照：「通信コマンドマクロツールの使用手順 (P.2-5)」	不要		必要	不要
ユニットマクロ処理項目を使うための準備 参照：「ユニットマクロ処理項目の使用手順 (P.2-5)」	不要			必要
設定画面の見方と設定方法 参照：「第 3 章 画面の見方と設定方法 (P.3-1)」	必要			
基本的なプログラムの書き方 参照：「第 4 章 プログラムの基本 (P.4-1)」	必要			
高度なプログラムの書き方 参照：「第 5 章 マクロプログラム (P.5-1)」	場合によって必要			
処理ユニットに関係するデータの種類 参照:「5-1-1 処理ユニットに関係するデータの種類 (P.5-2)」	場合によって必要			
システムに関係するデータの種類 参照：「5-1-2 システムに関係するデータの種類 (P.5-7)」	場合によって必要			
データのスコープと保存領域 参照：「5-1-3 データのスコープと保存領域 (P.5-9)」	場合によって必要			
状態遷移と実行タイミング 参照：「5-1-4 状態遷移と実行タイミング (P.5-12)」	不要	場合によって必要	不要	場合によって必要
処理の排他制御 参照：「5-1-5 処理の排他制御 (P.5-18)」	場合によって必要			
デバッグ機能の使い方 参照：「第 6 章 デバッグ機能 (P.6-1)」	必要			
トラブルシューティング 参照：「第 7 章 トラブルシューティング (P.7-1)」	場合によって必要			
マクロ関数一覧 参照：「第 8 章 マクロ関数 (P.8-1)」	場合によって必要			
マクロリファレンス一覧 参照：「第 9 章 マクロリファレンス一覧 (P.9-1)」	場合によって必要			

2

マクロカスタマイズ機能使用方法

2-1　マクロカスタマイズ機能の使い方について 2-2
2-1-1　マクロカスタマイズ機能の構成 2-2
2-1-2　マクロカスタマイズ機能の使用手順 2-3

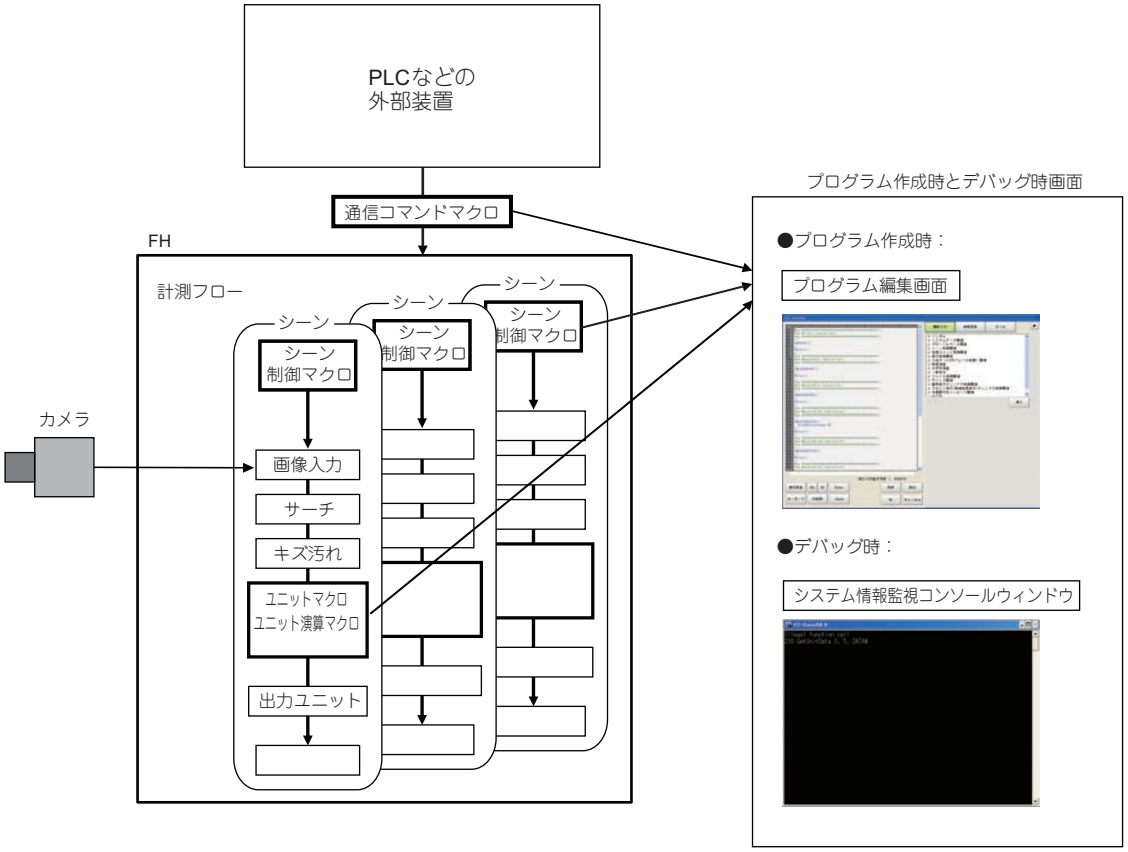
2-1 マクロカスタマイズ機能の使い方について

本稿参照の要否	ユニットマクロ	通信コマンドマクロ	シーン制御マクロ	ユニット演算マクロ
	必要			

マクロカスタマイズ機能は、補完・拡張したい処理をプログラミングすることで使用できます。プログラムの作成、実行、修正は、すべてセンサコントローラに用意されている機能で実施することができます。

2-1-1 マクロカスタマイズ機能の構成

マクロカスタマイズ機能は、処理項目やツールの設定画面のほか、各機能それぞれの設定画面でプログラムの作成や編集ができるプログラム編集画面と、プログラムが異常動作した場合にエラーの内容を確認できる、システム状態監視コンソールウィンドウで構成されています。



機能	説明
プログラム編集画面	マクロカスタマイズ機能の各機能の設定画面です。プログラム編集画面の内容は各機能で異なります。プログラム編集画面はプログラムを作成するときに使用します。プログラムの作成や編集と、入力補助機能が使用できます。 参照：「3-1-1 プログラム編集画面の見かた (P.3-2)」
システム状態監視コンソールウィンドウ	システムの状態がテキストで表示されるコンソールウィンドウです。マクロカスタマイズ機能で作成したプログラムが異常動作すると、システム状態監視コンソールウィンドウにエラー内容がテキストで表示されます。プログラムをデバッグするときに使用します。 参照：「3-1-2 システム状態監視コンソールウィンドウの見かた (P.3-6)」

使用上の注意

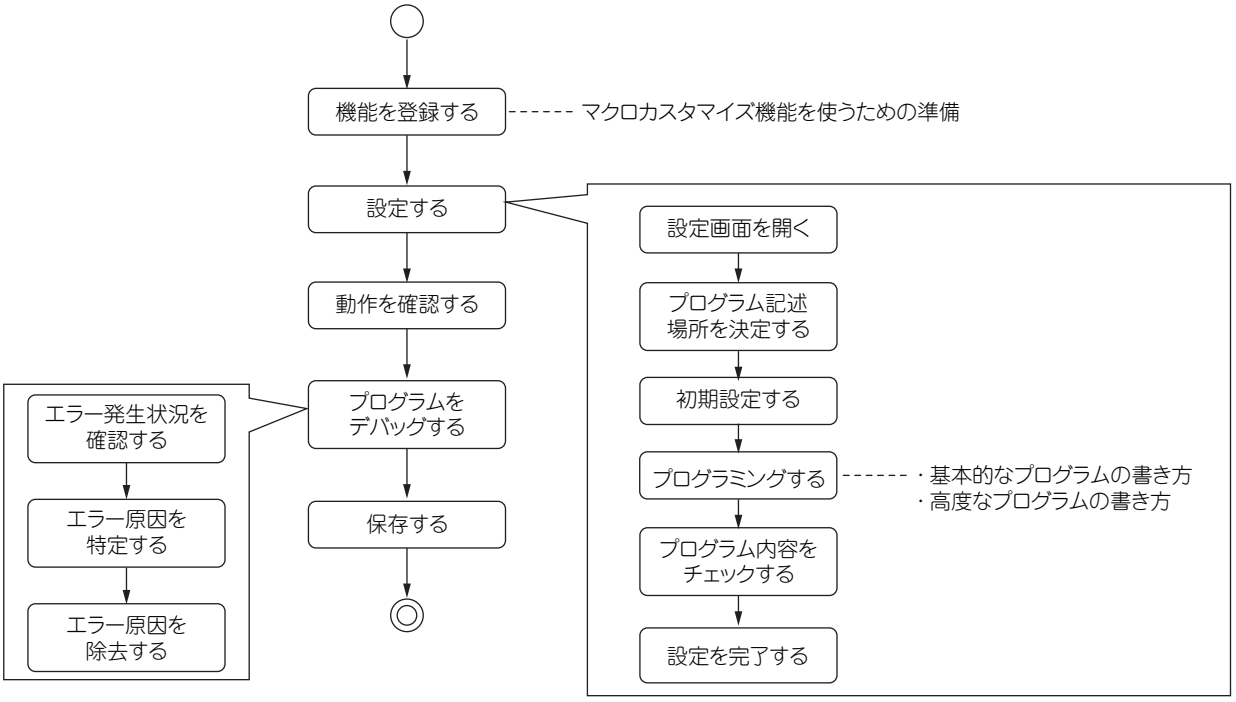
遠隔操作でマクロカスタマイズ機能の設定画面は表示できません。設定を変更する場合は、センサコントローラ上で直接、各機能の設定画面を開いてください。

2-1-2 マクロカスタマイズ機能の使用手順

マクロカスタマイズ機能を使用するための手順には、すべての機能で共通の使用手順と、機能ごとに個別の使用手順があります。

マクロカスタマイズ機能共通の使用手順

マクロカスタマイズ機能共通の使い方の流れは以下のとおりです。



項目	手順	説明
機能の登録、準備	機能の登録、設定画面の表示	マクロカスタマイズ機能が使用できる処理項目を計測フローに登録したり、ツール画面を起動したりします。具体的な使用手順は、使用する機能により異なります。
	↓	
機能の設定	機能の初期設定	初期値や変数を準備します。具体的な使用手順は、使用する機能により異なります。
	↓	
	処理内容の作成	処理させたい内容をプログラムで記述します。
	↓	
プログラムのデバッグ	動作確認	設定した内容が、期待通りに動作するか確認します。
	↓	
	デバッグ	プログラムが期待しない動作をする場合に実施します。期待しない動作の原因を特定して、期待通りの動作をするようにプログラムを修正します。
	↓	
保存	設定を保存	修正した内容を保存します。

ユニット演算マクロ処理項目の使用手順

ユニット演算マクロ処理項目の使い方の流れと、基本的な使用手順は以下のとおりです。

項目	手順	説明
登録、準備	ユニット演算マクロ処理項目の登録	計測フローにユニット演算マクロ処理項目を追加します。
	↓	
ユニット演算マクロ処理ユニットの設定（参照：「3-1-3 ユニット演算マクロ処理項目の設定画面の見かたと設定方法 (P.3-7)」）	使用する演算子の選択	ユニット演算マクロで使用する演算子にチェックを入れます。
	↓	
	参照変数の設定	参照変数を使用する場合に設定します。他の処理ユニットの外部参照データやシステムデータなど、当該ユニット演算マクロ処理項目以外のデータを参照変数として使用するための準備をします。
	↓	
	プログラムの入力	演算させたい内容をプログラムで記述します。
	↓	
	判定条件の設定	演算結果を判定するための条件を設定します。
	↓	
プログラムのデバッグ（参照：「6-1 デバッグ機能の使い方について (P.6-2)」）	動作確認	設定した内容が、期待通りに動作するか確認します。
	↓	
	デバッグ	処理ユニットが期待しない動作をする場合に実施します。期待しない動作の原因を特定して対策することで、ユニット演算マクロ処理ユニットが期待通りの動作をするようにプログラムを修正します。
	↓	
保存	設定を保存	修正した内容を保存します。

シーン制御マクロツールの使用手順

シーン制御マクロツールの使い方の流れと、基本的な使用手順は以下のとおりです。

項目	手順	説明
登録、準備	シーン制御マクロツールの起動	外部ツールからシーン制御マクロツールを起動します。
	↓	
シーン制御マクロツールの設定（参照：「3-1-4 シーン制御マクロツールの設定画面の見かたと設定方法 (P.3-12)」）	参照変数の設定	参照変数を使用する場合に設定します。処理ユニットの外部参照データやシステムデータなどを参照変数として使用するための準備をします。
	↓	
	ユニットラベルの設定	ユニットラベルを使用する場合に設定します。計測フロー内の処理ユニットを参照するときに、処理ユニット番号ではなくユニットラベルで参照できるようにするための準備をします。
	↓	
	プログラムの入力	シーンを制御したい内容をプログラムで記述します。
	↓	
プログラムのデバッグ（参照：「6-1 デバッグ機能の使い方について (P.6-2)」）	動作確認	設定した内容が、期待通りに動作するか確認します。
	↓	
	デバッグ	シーン制御マクロが期待しない動作をする場合に実施します。期待しない動作の原因を特定して対策することで、シーン制御マクロが期待通りの動作をするようにプログラムを修正します。
	↓	

項目	手順	説明
保存	設定を保存	修正した内容を保存します。

通信コマンドマクロツールの使用手順

通信コマンドマクロツールの使い方の流れと、基本的な使用手順は以下のとおりです。

項目	手順	説明
登録、準備	通信コマンドマクロツールの起動	外部ツールから通信コマンドマクロツールを起動します。
	↓	
通信コマンドマクロツールの設定（参照：「3-1-5 通信コマンドマクロツールの設定画面の見かたと設定方法 (P.3-13)」）	使用する通信コマンドマクロの選択	使用する通信コマンドマクロにチェックを入れます。
	↓	
	通信コマンドマクロ名の設定	通信コマンドマクロの名称を決定します。
	↓	
	計測中信号の出力の設定	処理中に BUSY 信号などの計測中信号を ON させるか OFF のままにするかを選択し、ON させたい場合にチェックを入れます。
	↓	
	プログラムの入力	処理させたい内容をプログラムで記述します。
	↓	
プログラムのデバッグ（参照：「6-1 デバッグ機能の使い方について (P.6-2)」）	動作確認	設定した内容が、期待通りに動作するか確認します。
	↓	
	デバッグ	通信コマンドマクロが期待しない動作をする場合に実施します。期待しない動作の原因を特定して対策することで、通信コマンドマクロが期待通りの動作をするようにプログラムを修正します。
	↓	
保存	設定を保存	修正した内容を保存します。

ユニットマクロ処理項目の使用手順

ユニットマクロ処理項目の使い方の流れと、基本的な使用手順は以下のとおりです。

項目	手順	説明
登録、準備	ユニットマクロ処理項目の登録	計測フローにユニットマクロ処理項目を追加します。
	↓	
ユニットマクロ処理項目の設定（参照：「3-1-6 ユニットマクロ処理項目の設定画面の見かたと設定方法 (P.3-20)」）	参照変数の設定	参照変数を使用する場合に設定します。他の処理ユニットの外部参照データやシステムデータなど、当該ユニットマクロ処理項目以外のデータを参照変数として使用するための準備をします。
	↓	
	プログラムの入力	処理させたい内容をプログラムで記述します。
	↓	
プログラムのデバッグ（参照：「6-1 デバッグ機能の使い方について (P.6-2)」）	動作確認	設定した内容が、期待通りに動作するか確認します。
	↓	
	デバッグ	ユニットマクロ処理ユニットが期待しない動作をする場合に実施します。期待しない動作の原因を特定して対策することで、ユニットマクロ処理ユニットが期待通りの動作をするようにプログラムを修正します。
	↓	
保存	設定を保存	修正した内容を保存します。

3

画面の見方と設定方法

3-1　画面の見かたと設定方法について 3-2
3-1-1　プログラム編集画面の見かた 3-2
3-1-2　システム状態監視コンソールウィンドウの見かた 3-6
3-1-3　ユニット演算マクロ処理項目の設定画面の見かたと設定方法 3-7
3-1-4　シーン制御マクロツールの設定画面の見かたと設定方法 3-12
3-1-5　通信コマンドマクロツールの設定画面の見かたと設定方法 3-13
3-1-6　ユニットマクロ処理項目の設定画面の見かたと設定方法 3-20
3-1-7　プログラムのセーブとロード 3-24

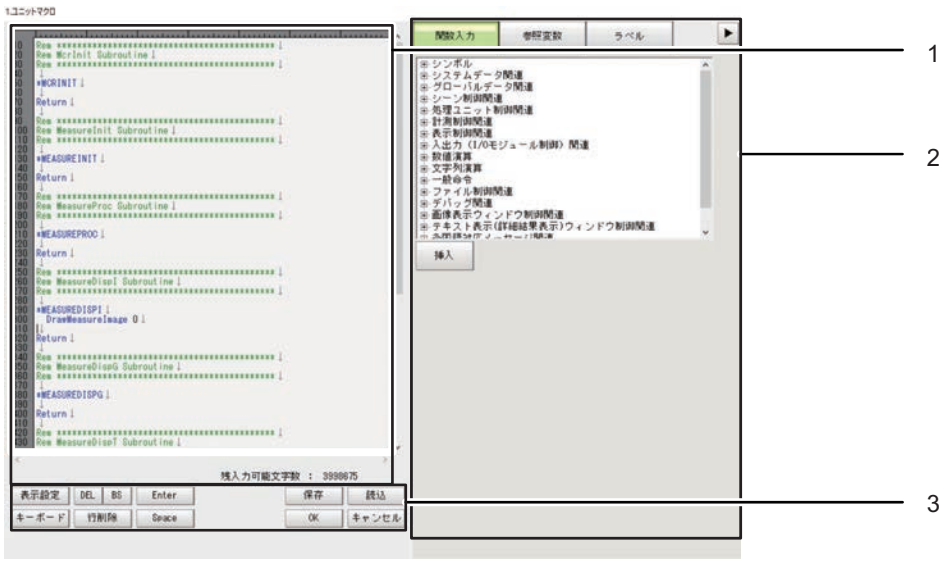
3-1　画面の見かたと設定方法について

本稿参照の要否	ユニットマクロ	通信コマンドマクロ	シーン制御マクロ	ユニット演算マクロ
	必要			

マクロカスタマイズ機能の設定画面には、マクロカスタマイズ機能のすべての機能で共通のプログラム編集画面と、各機能個別の設定画面があります。設定画面により設定方法も異なります。

3-1-1　プログラム編集画面の見かた

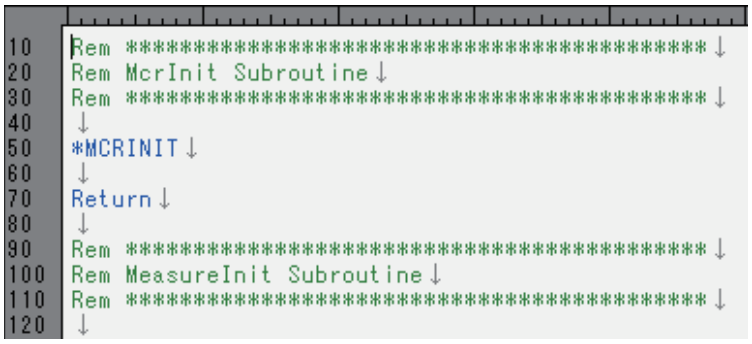
マクロカスタマイズ機能でプログラムを編集するときは、プログラム編集画面を使います。
プログラム編集画面は、以下のエリアで構成されています。



(1)**プログラムエリア**

プログラムが表示されるエリアです。表示される項目の一部は、表示するかどうかを選択することができます。詳細は、「表示設定 (P.3-5)」を参照してください。

- プログラム入力エリア
  プログラムが入力できるエリアです。このエリアでプログラムを作成、編集します。
  プログラム入力エリアで、プログラムを作成／編集するときは、キーボードを使用してプログラム入力エリアにプログラムを記述してください。
  スクリーンキーボードは、操作ボタンエリアから表示させることができます。詳細は、「(3) 操作ボタンエリア（P.3-4)」を参照してください。



プログラムエリアでは、制御文字が目に見える形で表示されます。制御文字はプログラムの残入力可能文字数のカウントに含まれるので、入力可能文字数を確認するときに便利です。詳細は、「残入力可能文字数表示エリア（P.3-4)」を参照してください。

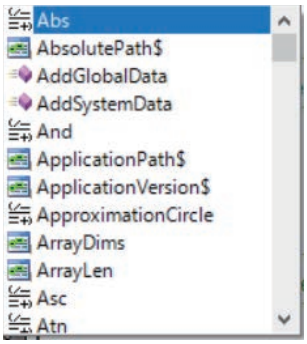
表示（色）	説明
↓（灰色）	改行文字を表します。
□（灰色）	全角空白文字を表します。
→（灰色）	タブ文字を表します。

使用上の注意

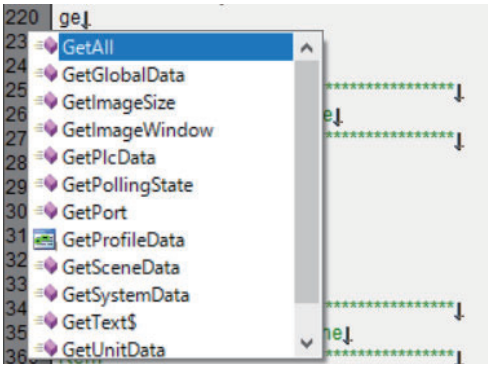
- プログラム入力エリアでは、注釈コメントが入力できます。
  設定できる文字列は、以下のとおりです。
  「Language setting」で設定した言語と英語が設定可能です。
  詳細は、「注釈コメント (P.4-4)」を参照してください。

参考

- プログラムに記述された関数のうち、使用できない関数は、関数名が赤字で表示されます。関数の使用可否は、使用するマクロカスタマイズ機能により異なります。詳細は、「第 9 章 マクロリファレンス一覧 (P.9-1)」を参照してください。
- 「Ctrl」＋「Space」または、「Alt」＋「→」キーを入力すると、候補の一覧が表示されます。候補の一覧から選択して、ダブルクリックまたは「Enter」キー入力で、プログラム入力エリアに入力することもできます。



- 2 文字以上入力すると、候補の一覧が自動的に表示されます。候補の一覧から選択して、ダブルクリックまたは「Enter」キー入力で、プログラム入力エリアに入力することもできます。



- 行番号表示エリア
  プログラムの行番号が表示されるエリアです。行番号はデバッグのときに使用します。行と行番号については、「4-1-1 基本構文 (P.4-2)」を参照してください。

参考

- 行番号にはプログラム内でユニークな番号が割り振られます。ひとつのユニット演算マクロ処理項目で複数の演算式を設定すると、行番号はひとつの処理項目内の複数の演算式で重複しないユニークな番号が割り振られます。

- 残入力可能文字数表示エリア
  プログラムであと何文字入力できるかが表示されるエリアです。改行文字やタブ文字などの制御文字も、文字数にカウントされます。プログラムは、残入力可能文字数が 0 文字以上になるように作成、編集してください。入力可能文字数を超えてプログラムの作成や編集をした場合は、プログラムが正常に動作しなくなる可能性があります。

(2)**プログラム入力補助エリア**

プログラム入力のための補助的な設定や入力補助操作ができるエリアです。

▶ボタンをクリックすると、プログラム入力補助エリアを非表示にする代わりに、プログラムエリアを横に広く表示できます。もう一度クリックすると、元の画面に戻ります。

- 関数入力タブ画面
  マクロ関数のリストが一覧表示されるエリアです。

ボタン	説明
挿入	プログラム入力エリアのカーソル位置直後に、選択したマクロ関数を挿入します。

- 参照変数タブ画面
  参照変数の登録と設定ができるエリアです。ユニット演算マクロ、シーン制御マクロ、ユニットマクロのプログラム編集画面でのみ表示されます。通信コマンドマクロでは表示されません。
  参照変数の詳細は、「4-1-3 変数 (P.4-6)」を参照してください。

ボタン	説明
挿入	プログラム入力エリアのカーソル位置直後に、選択した参照変数を挿入します。
追加	参照変数の設定画面が表示されます。設定画面から参照変数を追加できます。
削除	参照変数リストで選択した参照変数を削除します。
編集	参照変数リストで選択した参照変数の設定画面が表示されます。設定画面では参照変数の設定を変更できます。
保存	ファイルの保存画面が表示されます。保存先とファイル名を指定して、現在の参照変数リストの設定内容を XML 形式のファイルに保存します。
読込	ファイルの読込画面が表示されます。XML 形式でファイルに保存した参照変数を読み込みます。

- ラベルタブ画面
  ユニットラベルの設定と参照ができるエリアです。

ボタン	説明
挿入	プログラム入力エリアのカーソル位置直後に、選択したユニットラベルを挿入します。
編集	設定画面が表示されます。詳細は、「3-1-4 シーン制御マクロツールの設定画面の見かたと設定方法 (P.3-12)」を参照してください。

(3)**操作ボタンエリア**

プログラムをボタン操作で編集するためのボタンと、プログラムの保存と読込操作のボタンが用意されているエリアです。
操作できるボタンは、使用するマクロカスタマイズ機能によって異なります。
操作ボタンエリアで操作できるボタンは、以下のとおりです。

ボタン	説明
表示設定	表示設定画面を表示します。
キーボード	スクリーンキーボードを表示します。 プログラムを作成／編集する場合は、キーボードを使用してください。
DEL	プログラム入力エリアのカーソル位置直後の 1 文字を削除します。
BS	プログラム入力エリアのカーソル位置直前の 1 文字を削除します。
Enter	プログラム入力エリアのカーソル位置で改行します。
行削除	プログラム入力エリアのカーソル位置の行を削除します。

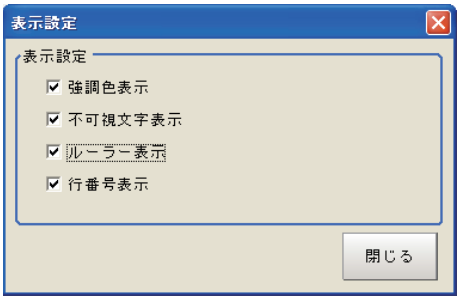
ボタン	説明
Space	プログラム入力エリアのカーソル位置に、スペースを挿入します。
クリア	プログラムエリアに表示されているプログラムを初期化します。 シーン制御マクロでのみ表示されます。
保存	プログラムファイルを保存します。ユニットマクロでのみ表示されます。
読込	プログラムファイルを読み込みます。ユニットマクロでのみ表示されます。
OK	プログラム編集画面での編集内容を確定して、画面を閉じます。 プログラムに誤りがある場合は、エラーダイアログが表示されます。エラーダイアログが表示される場合は、エラーを取り除いてから、OK をクリックしてください。エラーの内容は、システム状態監視コンソールウィンドウに表示されます。
キャンセル	プログラム編集画面での編集内容を破棄して、画面を閉じます。

参考

- FH シリーズでは、センサコントローラ上でのキー配列は、英語キーボードと同じ配列になります。日本語入力する場合は、［Alt］＋［~］で入力モードを切り替えることができます。
- ［OK］をクリックしたとき、プログラムに誤りがあると、エラーダイアログが表示されます。エラーダイアログが表示される場合は、エラーを取り除いてから OK をクリックすると、設定を完了することができます。エラーの内容は、システム状態監視コンソールウィンドウの表示で確認することができます。
  詳細は、「3-1-2 システム状態監視コンソールウィンドウの見かた (P.3-6)」を参照してください。

表示設定

プログラムエリアの表示内容を増やしてプログラムを分かりやすく表示したい場合には、プログラムエリアの表示設定を変更します。
表示設定画面の表示内容は、以下のとおりです。



設定項目	設定値 【工場出荷時】	説明
強調色表示	• 【チェックあり】 • チェックなし	プログラム入力エリアのプログラムを色分けして表示します。キーワードの種類によって色分けされるので、プログラムが見やすくなります。
不可視文字表示	• 【チェックあり】 • チェックなし	プログラム入力エリアで、改行文字やタブ文字などの制御文字を表示して目に見えるようにします。不可視文字の詳細は、「(1) プログラムエリア（P.3-2)」をご覧ください。
ルーラー表示	• 【チェックあり】 • チェックなし	プログラム入力エリアにルーラーを表示します。ルーラーを表示しておくことで、1 行の文字数を確認しやすくなります。
行番号表示	• 【チェックあり】 • チェックなし	プログラム入力エリアの左端に行番号を表示します。行番号を表示しておくことで、デバッグのときにエラーが発生した行を特定しやすくなります。

参考

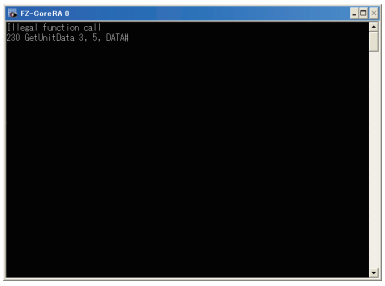
- 各表示設定のチェックを外すことで、プログラムエリアを広く表示させてプログラムを見やすくすることができます。また、プログラム編集操作のレスポンスを良くすることができます。
- 表示設定の内容は保存されません。プログラム編集画面を閉じると、表示設定の内容は工場出荷時の設定に戻ります。
- 強調色表示にチェックを入れた状態のとき、プログラムに記述された関数のうち使用できない関数は、関数名が赤字で表示されます。
  関数の使用可否は、使用するマクロカスタマイズ機能によって異なります。詳細は、「第 9 章 マクロリファレンス一覧 (P.9-1)」をご覧ください。

3-1-2 システム状態監視コンソールウィンドウの見かた

マクロカスタマイズ機能でプログラムのエラー内容を確認したりデバッグしたりするときは、システム状態監視コンソールウィンドウを使います。

システム状態監視コンソールウィンドウの構成

システム状態監視コンソールウィンドウの表示内容は、以下のとおりです。



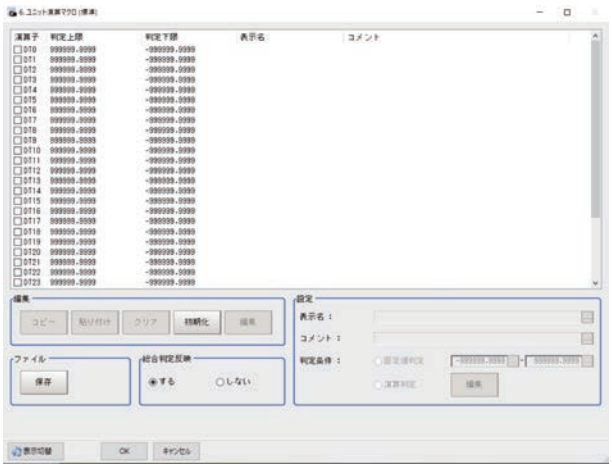
表示エリア	説明
システム状態表示エリア	システムの状態がテキストで表示されます。 マクロカスタマイズ機能で作成したプログラムが異常動作すると、システム状態表示エリアにエラー内容が表示されます。 表示される内容は、「システム状態監視コンソールウィンドウの確認 (P.6-4)」を参照してください。

使用上の注意

- システム状態監視コンソールウィンドウの右上にある［×］ボタンをクリックするなどの方法で、システム状態監視コンソールウィンドウを終了しないでください。システムが正常に動作しなくなる可能性があります。もし、システム状態監視コンソールウィンドウを終了してしまった場合は、設定を保存し、センサコントローラを再起動してください。
- システム状態監視コンソールウィンドウにテキストを表示する場合、システム状態監視コンソールウィンドウを最小化していると、処理に時間がかかります。

3-1-3　ユニット演算マクロ処理項目の設定画面の見かたと設定方法

「ユニット演算マクロ」処理項目のプロパティ設定画面の構成は、以下のとおりです。



使用する演算子の選択（ユニット演算マクロ）

ひとつのユニットで設定できる演算処理の数は、「DT0」～「DT31」の 32 です。
「DT0」から昇順に処理が実行されます。

参考

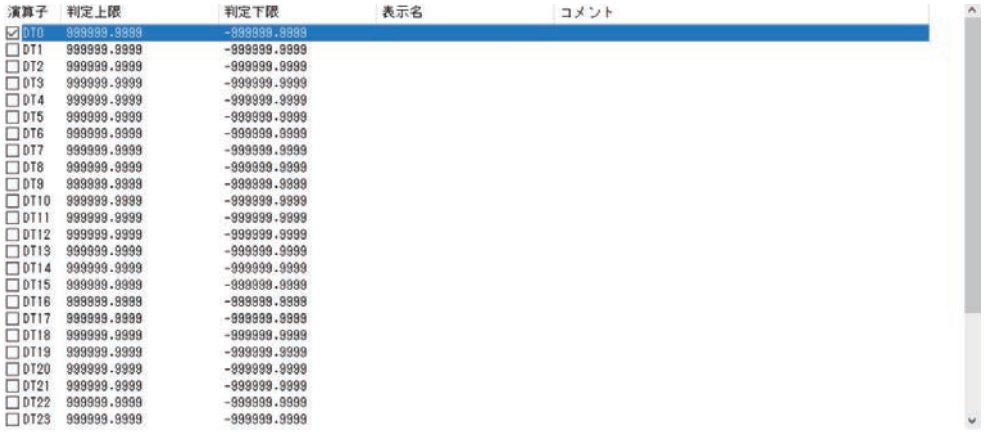
ユニット演算マクロを設定しただけでは、演算結果は外部装置に出力されません。演算結果を外部装置に出力する場合は、フロー編集で「ユニット演算マクロ」以降のユニットに結果出力関連の処理項目を設定してください。詳細は、『画像処理システム FH/FHV シリーズ処理項目機能リファレンスマニュアル（SDNB-713)』の「結果を外部へ出力する」を参照してください。

1 **一覧から、演算処理を設定する演算子をクリックします。**

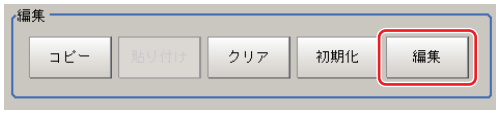


演算子	判定上限	判定下限	表示名	コメント
DT0	999999.9999	-999999.9999		
DT1	999999.9999	-999999.9999		
DT2	999999.9999	-999999.9999		
DT3	999999.9999	-999999.9999		
DT4	999999.9999	-999999.9999		
DT5	999999.9999	-999999.9999		
DT6	999999.9999	-999999.9999		
DT7	999999.9999	-999999.9999		
DT8	999999.9999	-999999.9999		
DT9	999999.9999	-999999.9999		
DT10	999999.9999	-999999.9999		
DT11	999999.9999	-999999.9999		
DT12	999999.9999	-999999.9999		
DT13	999999.9999	-999999.9999		
DT14	999999.9999	-999999.9999		
DT15	999999.9999	-999999.9999		
DT16	999999.9999	-999999.9999		
DT17	999999.9999	-999999.9999		
DT18	999999.9999	-999999.9999		
DT19	999999.9999	-999999.9999		
DT20	999999.9999	-999999.9999		
DT21	999999.9999	-999999.9999		
DT22	999999.9999	-999999.9999		
DT23	999999.9999	-999999.9999		

2 **演算処理を実行する演算子にチェックを入れます。**



3 **編集エリアで、[編集] をクリックします。**



ユニット演算マクロの設定画面が表示されます。

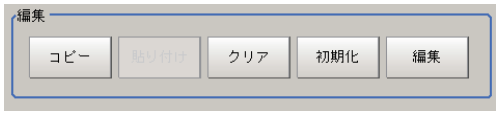
演算子の編集（ユニット演算マクロ）

設定した演算処理の設定をコピーしたりクリアしたりすることができます。

1 **一覧から、演算処理の設定を編集する演算子をクリックします。**



2 **編集エリアで、各操作を選択します。**



ユニット演算マクロの設定画面が表示されます。

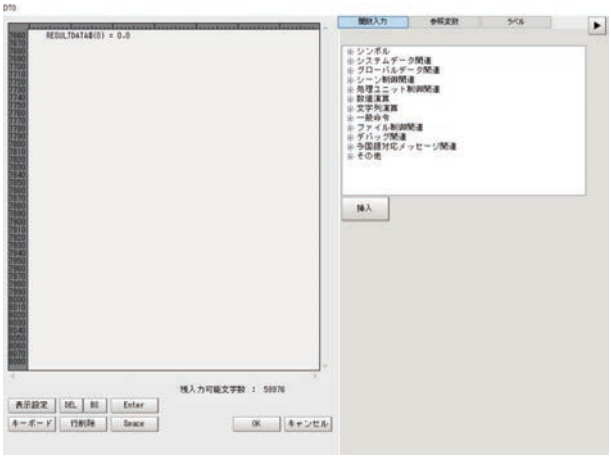
項目	説明
コピー	一覧で選択した演算子の設定をコピーします。コピーした設定内容は、他の演算子に貼り付けできます。
貼り付け	一覧で選択した演算子に、コピーした設定値を貼り付けます。 貼り付けられるデータは、有効無効フラグ、演算マクロコード、表示名、コメント、判定フラグ、固定判定上下限、判定マクロコードです。

項目	説明
クリア	一覧で選択した演算子の設定を初期化します。初期化対象のデータは、有効無効フラグ、演算マクロコード、表示名、コメント、判定フラグ、固定判定上下限、判定マクロコードです。
初期化	すべての演算子の設定を初期化します。
編集	一覧で選択した演算子の設定を編集します。

参照変数の設定（ユニット演算マクロ）

プログラムで使用する参照変数を設定します。設定方法は「ユニットマクロ」処理項目と同じです。
詳細は、「参照変数の設定（ユニットマクロ）（P.3-20)」を参照してください。

プログラムの入力（ユニット演算マクロ）



注:RESULTDATA#() は演算結果のデータ識別名です。

プログラムの入力方法は「ユニットマクロ」処理項目と同じです。
詳細は、「プログラムの入力（ユニットマクロ）（P.3-22)」を参照してください。

判定条件の設定（ユニット演算マクロ）

1 **設定エリアで、判定条件を設定します。**



設定項目	設定値 【工場出荷時】	説明
表示名	-	一覧で選択した演算処理の表示名を入力します。表示名は詳細結果エリアにも表示されます。表示名が空欄の場合は、詳細結果エリアには「演算 N」（N=0～31）として表示されます。 言語識別子を使った多言語書式で指定できます。言語識別子を使った記載の方法については、『画像処理システム FH/FHV シリーズ ユーザーズマニュアル（SDNB-719)』の「文字を入力する」を参照してください。
コメント	-	一覧で選択した演算処理のコメントを入力します。

設定項目	設定値 【工場出荷時】	説明
判定条件	• 【固定値判定】 • 演算判定	演算結果の判定を固定値かマクロ判定かを選択します。固定値判定を選択した場合は、OK とする上下限値を設定します。マクロ判定を選択した場合は、［編集］をクリックして、算出した値に対する演算処理を定義します。

2 **総合判定反映エリアで、シーンの総合判定に反映するかを選択します。**



設定項目	設定値 【工場出荷時】	説明
総合判定反映	• 【する】 • しない	この処理ユニットの判定結果を、シーンの総合判定に反映するか選択します。

テスト計測と調整のポイント（ユニット演算マクロ）

計測の精度や速度を高めるために、テスト計測して計測結果を確認しながらパラメータを調整します。詳細結果エリアには、テキストで次の内容が表示されます。

表示項目	説明
判定	判定結果
演算 0 の表示名	演算 0 の値
演算 1 の表示名	演算 1 の値
:	:
演算 31 の表示名	演算 31 の値

画像表示エリアに表示する画像は、画像表示設定のサブ画像で指定した画像が表示されます。

サブ画像番号	表示される画像の説明
0	計測画像

調整のポイント（ユニット演算マクロ）

次のポイントを参考にして調整方法を決めてください。

● コンソールウィンドウにエラーメッセージが出る

調整すべきパラメータ	処置
-	「9-1 エラー一覧 (P.9-2)」を参照してください。

●「DEL」、「BS」、「Enter」などをクリックしても動作しない

調整すべきパラメータ	処置
-	コードウィンドウにフォーカスが無い（キー入力カーソルが表示されていない）状態では動作しません。操作したい位置をクリックした上で、ボタンをクリックしてください。

● 文字列に改行コードを含めたい

調整すべきパラメータ	処置
マクロコード	CR → Chr$(13)LF → Chr$(10) を文字列に加算 (+) してください。

● 演算結果が未計測になる

調整すべきパラメータ	処置
有効／無効	演算子にチェックを入れて有効にしてください。
演算判定	演算判定において、正しく判定結果が設定されていない可能性があります。

計測結果の表示や出力内容は、「出力できる計測結果（ユニット演算マクロ）（P.3-11)」を確認してください。

出力できる計測結果（ユニット演算マクロ）

ユニット演算マクロの計測結果は、以下のとおりです。計測結果は、詳細結果エリアに表示されるほか、結果出力関連の処理項目を使うことで、外部装置へ出力できます。
計測結果も含めた参照可能なパラメータの一覧については、「外部参照テーブル（ユニット演算マクロ）（P.3-11)」をご覧ください。

計測項目	文字列	内容
判定	JG	判定結果
演算結果 0	DT00	演算結果 0
判定結果 0	JG00	判定結果 0
:	:	:
演算結果 31	DT31	演算結果 31
判定結果 31	JG31	判定結果 31

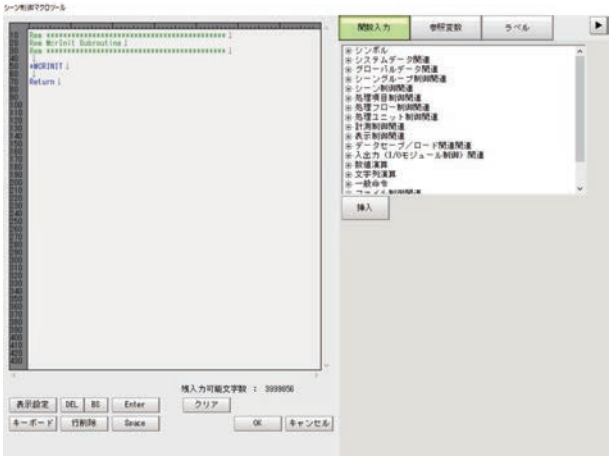
外部参照テーブル（ユニット演算マクロ）

処理ユニットデータ設定／取得機能を持つ処理項目や制御コマンドから番号を指定することで、次のデータを参照できます。

番号	データ名	設定／取得	データ範囲
0	判定	取得のみ	0：判定なし（未計測） 1：判定結果 OK -1：判定結果 NG
5 + N × 1 (N = 0 ～ 31)	演算結果 N (N = 0 ～ 31)	設定／取得	-99999.9999 ～ 99999.9999
37 + N × 1 (N = 0 ～ 31)	判定結果 N (N = 0 ～ 31)	取得のみ	0：判定なし（未計測） 1：判定結果 OK -1：判定結果 NG

3-1-4　シーン制御マクロツールの設定画面の見かたと設定方法

シーン制御マクロツールの設定画面は、プログラム編集画面と同じです。設定画面の詳細は、「3-1 画面の見かたと設定方法について (P.3-2)」を参照してください。



参照変数の設定（シーン制御マクロ）

プログラムで使用する参照変数を設定します。設定方法は、「ユニットマクロ」処理項目と同じです。詳細は、「参照変数の設定（ユニットマクロ）(P.3-20)」を参照してください。

ユニットラベルの設定（シーン制御マクロ）

ユニットラベルを設定すると、プログラムの中で、処理ユニットを処理ユニット番号ではなくラベルで指定することができます。処理ユニットをラベルで指定するためには、ユニットラベルと Ut 関数を使用します。ユニットラベルを使用することで、計測フローを変更して処理ユニット番号が変わっても、プログラムを変える必要がなくなります。
設定したユニットラベルは、他のマクロカスタマイズ機能で参照することが可能です。
詳細は、マクロ関数リファレンスの Ut 関数「Ut (P.8-320)」を参照してください。

1 **シーン制御マクロツールの設定画面で、［ラベル］タブをクリックします。**
現在のシーンで設定されているユニットラベルの一覧が表示されます。

2 **［編集］をクリックします。**
ユニットラベル画面が表示されます。

3 **ユニットラベルを設定したい処理ユニットを選択します。**

4 **…をクリックして、ユニットラベル名を設定します。**

設定項目	設定値 【工場出荷時】	説明
ユニットラベル	1 ～ 32 文字	ユニットラベル名を設定します。ここで設定したユニットラベルは、Ut 関数で使用します。 ユニットラベル名には、アルファベット、数字、”.”、“_” が使用できます。

5 **［OK］をクリックします。**

使用上の注意

- ユニットラベルの設定は、シーンごとにシーンデータに保存されます。ユニットラベルの設定を別のシーンでも使用したい場合は、別途設定し直すか、シーンメンテナンス機能でシーンをコピーして使用してください。詳細は、『画像処理システム FH/FHV シリーズ ユーザーズマニュアル（SDNB-719)』の「シーンを編集する」を参照してください。
- 同じシーン内で、重複するユニットラベルを設定することはできません。

プログラムの入力（シーン制御マクロ）

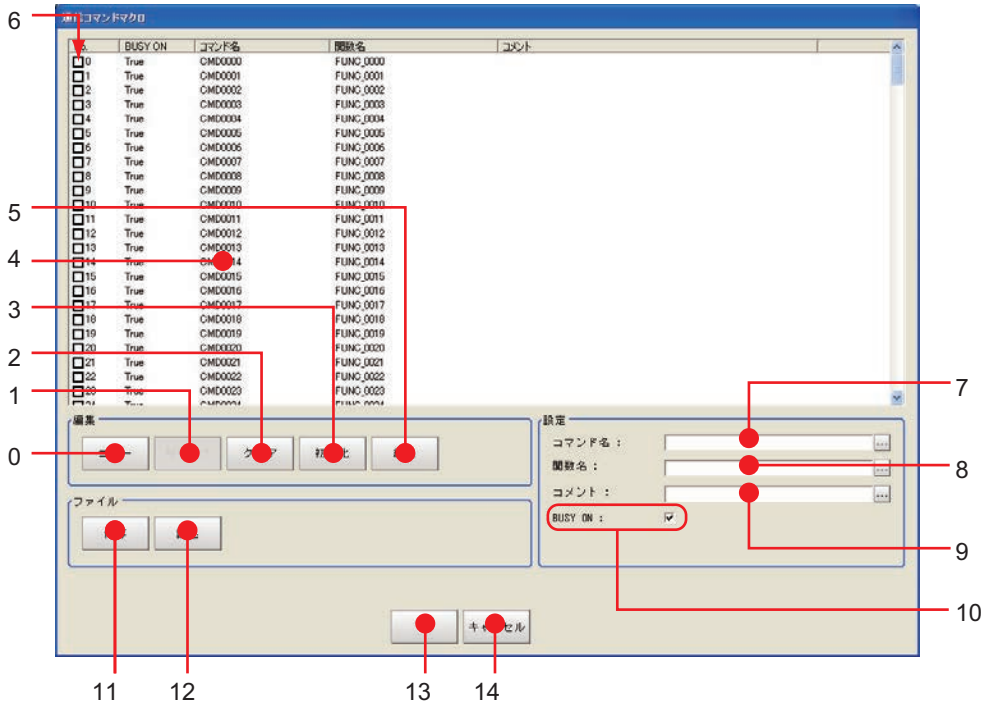
プログラムの入力方法は、「ユニットマクロ」処理項目と同じです。
詳細は、「プログラムの入力（ユニットマクロ）（P.3-22)」を参照してください。

3-1-5　通信コマンドマクロツールの設定画面の見かたと設定方法

通信コマンドマクロツールの設定画面の構成は、以下のとおりです。

通信コマンドマクロの設定手順

通信コマンドマクロの作成／編集ができます。
最大 256 個（No.0 ～ 255）の通信コマンドマクロを作成できます。



● 画面の説明

No.	名称	内容
0	コピー	コマンドリストで選択中のコマンド情報をコピーします。
1	貼り付け	コピーされている情報をコマンドリストで選択中のコマンドに上書きします。貼り付け対象は「コメント」(処理内容）です。また、コピーされていない状態では無効化されています。
2	クリア	コマンドリストで選択中のコマンドを初期化します。
3	初期化	すべてのコマンドを初期化します。

No.	名称	内容
4	コマンドリスト	コマンド一覧を表示します。
5	編集	コマンドリストで選択中のコマンドの処理内容を編集します。
6	有効／無効	カスタムコマンドを有効にするか設定／表示します。 定義されていてもチェックがされていないと実行されません。
7	コマンド名	リストで選択中のコマンドのコマンド名を設定／表示します。
8	関数名	リストで選択中のコマンドの関数名を設定／表示します。
9	コメント	リストで選択中のコマンドのコメントを設定／表示します。言語識別子を使った多言語書式で指定できます。言語識別子を使った記載の方法については、『画像処理システム FH/FHV シリーズ ユーザーズマニュアル (SDNB-719)』の「文字を入力する」を参照してください。
10	Busy On	コマンド実行前に計測禁止状態 (MeasureStop) にするかを設定／表示します。チェックされているとコマンド実行中は BUSY が ON され、コマンド実行後に計測禁止状態が解除されます (MeasureStart)。その後、MeasureInit イベントが発行されます。
11	保存	すべての通信カスタムコマンドをファイルに保存します。
12	読込	ファイルから通信カスタムコマンドを読み込みます。 既存のデータは上書きされます。
13	OK	内容を保存して終了します。
14	キャンセル	内容を保存せずに終了します。

使用する通信コマンドマクロの選択（通信コマンドマクロ）

定義したいコマンド No. 列をチェックします。チェックをいれたコマンド No. のみが有効となります。

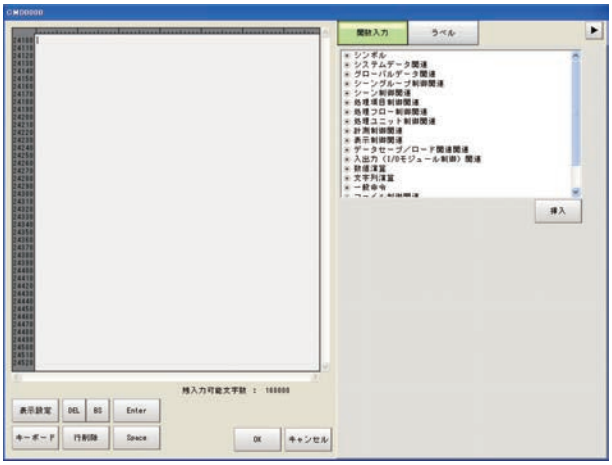
通信コマンドマクロ名の設定（通信コマンドマクロ）

通信コマンドマクロ画面右下の設定エリアの「コマンド名」で、無手順のときに使用する「コマンド名」を入力します。

計測中信号の出力の設定（通信コマンドマクロ）

追加カスタムコマンド画面右下の設定エリアの「BUSY ON」で、設定している通信カスタムコマンドの実行中に、BUSY を ON(True) にするか、OFF(False) にするかを指定します。

プログラムの入力（通信コマンドマクロ）



プログラムの入力方法は、「ユニットマクロ」処理項目と同じです。
詳細は、「プログラムの入力（ユニットマクロ）（P.3-22)」を参照してください。

通信コマンドマクロの指定方法

作成した通信カスタムコマンドは、外部からセンサコントローラに対して発行します。通信プロトコルに応じて、以下のように異なります。

● 無手順の場合

コマンド名の文字列を、ASCII 文字列で指定します。
「コマンド名」 + 半角スペース + 「数値」の ASCII 文字列を発行します。
標準コマンドも使う場合は、重複しないコマンド名にしてください。
例：コマンド名「mycommand」で、引数に数値 18 を指定する場合
「mycommand 18」を発行します。

● パラレルの場合

コマンド No. の 0 ～ 255 を、2 進数で指定します。DI0 ～ DI6 の 7 ビットに、コマンド No. を指定します。その上で、実行ビットの DI7 を OFF → ON にします。
ただし、コマンドは 7 ビットの制限があるため、コマンド No .が「0 ～ 127」のときと、128 ～ 255 のときで、指定方法が以下のように異なります。

- **0 ～ 127 のとき：DI0 ～ DI6 の 7 ビットをそのまま指定します。**
  例：コマンド No. 120 のとき、120 は 2 進数では 111 1000 のため、DI6: 1、DI5: 1、DI4: 1、DI3: 1、DI2: 0、DI1: 0、DI0: 0 で指定した後、1 [ms] 以上後に DI7 を OFF→ ON することでコマンドを発行します。
- **128 ～ 255 のとき：DI0 ～ DI6 の 7 ビット指定 0 ～ 127 で不足する分を、端子オフセットコマンド「DIOFFSET」であらかじめ加算します。**
  例：コマンド No. 150 のとき、150 は 2 進数では 1001 0110 のため、8 ビットとなります。そのため、たとえば、DI0 ～ DI6 は、150 の半分の 75 として、残りの 75 を端子オフセットコマンド DIOFFSET であらかじめ加算します。以下の手順です。

1 **DIOFFSET 75 を発行します。**

2 **75 は 2 進数では 100 1011 のため、DI6: 1、DI5: 0、DI4: 0、DI3: 1、DI2: 0、DI1: 1、DI0: 1 で指定した後、1 [ms] 以上後に DI7 を OFF → ON することでコマンドを発行します。**

3 **DIOFFSET 0 を発行します。**
（0: OFF 状態、1: ON 状態）

- **無手順、パラレル以外（PLC リンク、EtherCAT、EtherNet/IP、PROFINET）の場合**

  コマンドコード(CMD-CODE) として、コマンド No.の 0 ～ 255 を 16 進数 00 ～ FFHex で指定します。配置順は、16 進数の上位桁→下位桁を、PLC の I/O メモリの上位 CH アドレス→下位（若い）CH アドレスの順に指定します。この場合、FFHex が最大のため、命令エリアの+2 のみに 00 ～ FFHex を指定し、命令エリアの+3 には 0000Hex を指定します。
  例：コマンド No. 120 のとき、16 進数で 0078Hex のため、命令エリアの+2: 0078Hex、命令エリアの+3: 0000Hex を指定します。

通信コマンドマクロの動作 共通事項

基本の流れ
一般的には、各コマンドは以下のような流れで処理を行います。
(1)入力コマンド・パラメータが正しいかのチェック（範囲、型など）
(2)処理本体
(3)処理結果・レスポンスの出力
IO モジュールの種類（文字列、パラレル、…）によって、入出力のコマンド・パラメータ、レスポンスの返し方は異なります。
詳細については、以下のページを参照してください。

- 「無手順コマンドの作り方 (P.3-18)」
- 「パラレルコマンドの作り方 (P.3-19)」
- 「PLC リンクコマンドの作り方 (P.3-19)」
- 「フィールドバスコマンドの作り方 (P.3-20)」

BUSY 信号の処理

コマンドマクロ一覧画面の BUSY フラグは、基本的に ON で運用してください。

使用上の注意

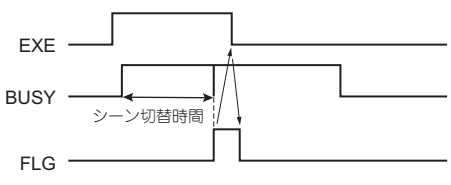
BUSY フラグが ON の状態で計測を実行（Measure コマンド）するとエラーが発生します。
BUSY フラグ ON が必要なコマンドと組み合わせたい場合、たとえばシーン切替と計測を連続して実行させたい場合は、BUSY フラグは OFF にした状態で、コマンド処理内で下記のように設定してください。

MeasureStop	'Busy を ON にする（計測禁止）
CangeSceneArgumentValue#(0)	'BusyOn（計測禁止）が必要なコマンドを実行する
MeasureStart	'計測の前に Busy を OFF にする（計測許可）
Measure	'計測を実行する

通信コマンドマクロ実行時の BUSY のタイミングは、「BUSY ON」のチェックを外した場合、通信マニュアル記載のタイミングと異なる場合があります。
シーン切替と計測を連続して実行する場合のタイミング例を示します。
MeasureStart/MeasureStop コマンドによりタイミングは次のように異なります。
処理中は BUSY ON とするためには、Measure Stop、Measure Start コマンドを入れてください。

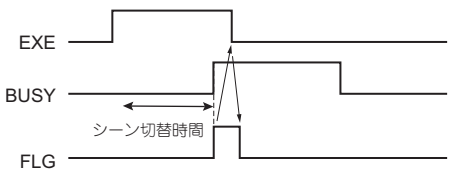
(1)通信コマンドマクロ内で計測トリガの入力を禁止し、シーン切り替え、計測トリガの入力を許可、Measure を実行する場合

```
Measure Stop↓
ChangeScene ArgumentValue#(0)↓
Measure Start↓
Measure↓
```



(2)通信コマンドマクロ内でシーン切り替え、Measure を実行する場合

```
ChangeScene ArgumentValue#(0)↓
Measure↓
```



標準コマンドとの関連

標準のコマンドと同じコマンド名／コマンド番号の処理を通信コマンドマクロで作成した場合、通信コマンドマクロが優先され、標準のコマンドは実行されません。
カスタムコマンド処理後に標準搭載のコマンド処理も実行したい場合は、以下を設定してください。

```
CommandExecute&=False
```

上記処理を実行した場合、通信コマンドマクロの処理が実行された後、標準コマンドの処理が実行されます。

別コマンドの処理の呼び出し

コマンド処理中に他のコマンドで定義した処理を呼び出すことができます。
各コマンドには、「関数名」が定義されており、この関数名を用いて処理を呼び出します。

例：下記の表のようにコマンドが定義されている場合

コマンド No.	コマンド名	Busy	コマンド関数名
0	CMD0	False	FUNC0
1	CMD1	True	FUNC1

CMD1 の処理の中で、CMD0 の処理を呼び出したい場合

```
Gosub *FUNC0
```

と記述すると呼び出すことができます。
この際、BUSY の動作などは呼び出し元の設定に準じるため、BUSY は処理が完了するまで ON のまま（CMD1 での設定）です。

使用上の注意

お互いの処理を相互に呼び出すような記述を行なった場合、上記例では CMD0 に CMD1 の処理を呼び出す記述が記載されていた場合、無限ループに陥って動作しなくなりますので注意してください。

通信モジュールごとに動作を分けたい場合

変数 Ioldent$ には、コマンドを入力した IO モジュールの識別名がセットされています。
通信モジュール毎に動作を分けたい場合、Ioldent$ の値によって処理を分けてください。

例：シリアル無手順で受信した場合は応答に「Serial」を、UDP 無手順で受信した場合は応答に「Ethernet」を返すコマンド

```
If Ioldent$ = "SerialNormal" Then
   Response$ = "Serial"
Elseif Ioldent$ = "UdpNormal" Then
   Response$ = "Ethernet"
Endif
```

無手順コマンドの作り方

- **コマンド・パラメータ**
  入力された無手順コマンドは、半角スペース" " でコマンド・パラメータが区切られているものとして解釈され、あらかじめ定義された次の変数に内容がセットされます。

変数名	型	内容
ArgumentsLength&	整数	パラメータの数（0 ～ 32）
ArgumentString$()	文字列配列	パラメータの配列（文字列）
ArgumentValue#()	実数配列	数値変換したパラメータの配列 注：変換できない場合は 0 がセットされます。

```
"AAA param0 param1 param2"
```

上記の文字列を受信した場合、次のように設定されます。

```
ArgumentsLength&      : 3（パラメータの数）
ArgumentString$(0)    : param0（文字列型）
ArgumentString$(1)    : param1（文字列型）
ArgumentString$(2)    : param2（文字列型）
ArgumentValue#(0)     : param0 を数値に変換したもの（変換できない場合は 0）
ArgumentValue#(1)     : param1 を数値に変換したもの（変換できない場合は 0）
ArgumentValue#(2)     : param2 を数値に変換したもの（変換できない場合は 0）
```

例：「SC1」を受信してシーン 1 に切り替えるコマンドを作成する場合

```
SceneChange ArgumentValue#(0)
```

- **レスポンス出力**
  次の変数に値をセットすることで、コマンド処理の結果を返すことができます。

変数名	型	内容
ResponseString$	文字列	戻り値
ResponseCode&	整数	コマンド実行結果 • 0：コマンド処理の成功（「OK」を返す） • -1：コマンド処理の失敗（「ER」を返す） • -2：無応答（「OK/ER」を返さない）

使用上の注意

ResponseCode& が未設定の場合：コマンド実行結果は 0 を返します。エラーが発生した場合は-1 を返します。

例：「TEST」コマンドに以下のプログラムをセットします。

```
ResponseString$ = "TestString"
```

コマンドとレスポンスは以下のようになります。

```
-> TEST
<- TestString
<- OK
```

パラレルコマンドの作り方

- **レスポンス出力**

  次の変数に値をセットすることで、コマンド処理の結果を返すことができます。

変数名	型	内容
ResponseCode&	整数	コマンド実行結果 • 0：コマンド処理の成功 • -1：コマンド処理の失敗（ERROR 信号が ON します。） • -2：無応答

使用上の注意

ResponseCode& が未設定の場合：コマンド実行結果は 0 を返します。エラーが発生した場合は-1 を返します。

PLC リンクコマンドの作り方

- **コマンド・パラメータ**

  あらかじめ定義された次の変数に内容がセットされます。

変数名	型	内容
ArgumentsLength&	整数	パラメータの数（0 ～ 6）
ArgumentValue#()	実数配列	パラメータの配列 注：2ch 分のデータ（整数型）

- **レスポンス出力**

  次の変数に値をセットすることで、コマンド処理の結果を返すことができます。

変数名	型	内容
ResponseValue&()	配列	戻り値
ResponseCode&	整数	コマンド実行結果 • 0：コマンド処理の成功 • -1：コマンド処理の失敗 • -2：無応答

使用上の注意

ResponseCode& が未設定の場合：コマンド実行結果は 0 を返します。エラーが発生した場合は-1 を返します。

フィールドバスコマンドの作り方

- **コマンド・パラメータ**
  あらかじめ定義された次の変数に内容がセットされます。

変数名	型	内容
ArgumentsLength&	整数	パラメータの数（0 ～ 3）
ArgumentValue#()	実数配列	パラメータの配列 注：2ch 分のデータ（整数型）

- **レスポンス出力**
  次の変数に値をセットすることで、コマンド処理の結果を返すことができます。

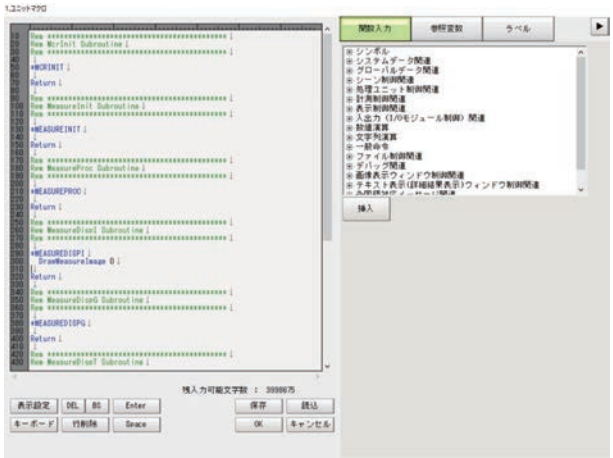
変数名	型	内容
ResponseValue&()	配列	戻り値
ResponseCode&	整数	コマンド実行結果 • 0：コマンド処理の成功 • -1：コマンド処理の失敗 • -2：無応答

使用上の注意

ResponseCode& が未設定の場合：コマンド実行結果は 0 を返します。エラーが発生した場合は-1 を返します。

3-1-6　ユニットマクロ処理項目の設定画面の見かたと設定方法

「ユニットマクロ」処理項目のプロパティ設定画面は、プログラム編集画面と同じです。
詳細は、「3-1-1 プログラム編集画面の見かた (P.3-2)」を参照してください。

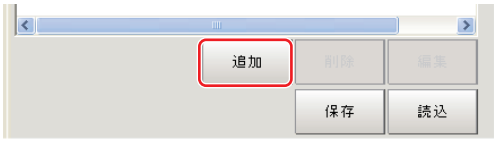


参照変数の設定（ユニットマクロ）

関数で使用する参照変数を設定します。

1 **マクロの設定画面で、参照変数リストの［追加］をクリックします。**

参照変数画面が表示されます。



2 **…をクリックして、変数名を設定します。**

変数名は、先頭大文字で英数字のみが設定可能です。

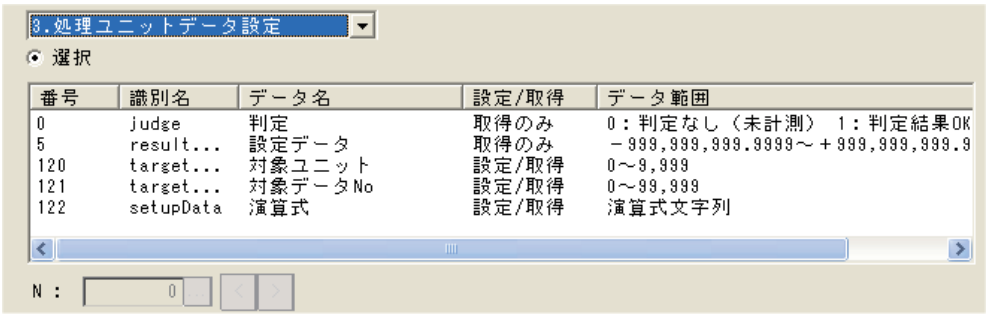


3 **参照する変数を設定します。**



- 「ユニット」を選択した場合

  参照する処理項目を選択し、一覧から参照するデータを選択します。



- 「グローバル」を選択した場合

  …をクリックして、グローバルデータの識別名を設定します。グローバルデータの識別名は、AddGlobalData 関数で定義します。

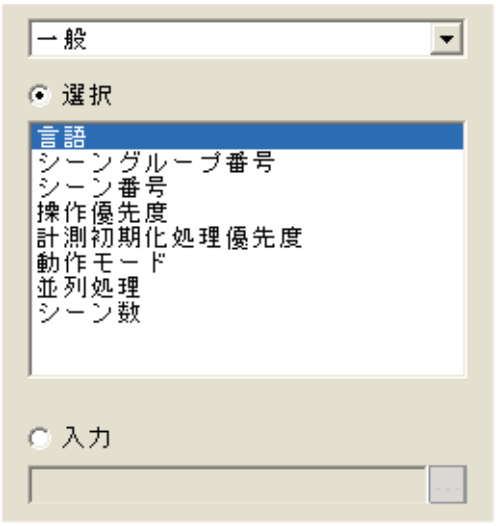
  詳細は、「AddGlobalData (P.8-23)」を参照してください。



- 「システム」を選択した場合

  ［選択］にチェックを入れて、システム変数の種類を選択した後、一覧から参照する変数を選択します。

  変数値を設定する場合は、［入力］にチェックを入れた後、…をクリックして、変数値を設定します。



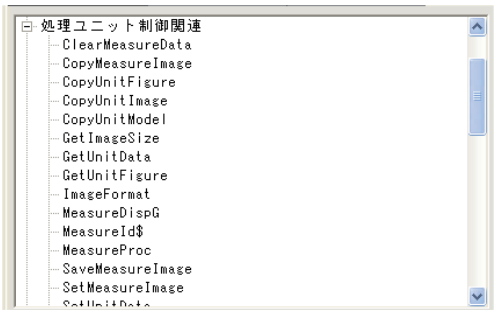
4 **[OK] をクリックします。**

ユニットラベルの設定（ユニットマクロ）

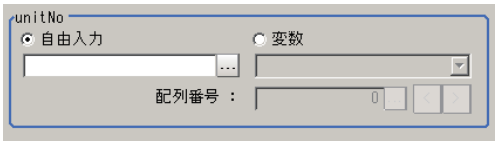
プログラムで使用するユニットラベルを設定します。設定方法は、「シーン制御マクロ」と同じです。詳細は、「ユニットラベルの設定（シーン制御マクロ）（P.3-12)」を参照してください。

プログラムの入力（ユニットマクロ）

1 **ユニット演算マクロの設定画面で、関数リストで挿入する関数を選択します。**
関数を選択すると、関数リストの下に引数リストが表示されます。



2 **引数を設定します。**



設定項目	設定値 【工場出荷時】	説明
引数入力方法	• 【自由入力】 • 変数	• 自由入力 引数を自由に入力する場合に選択します。 …をクリックして、引数を設定します。 • 変数 引数を変数から選択する場合に選択します。 ▼をクリックして、変数を選択します。 マクロコード上で定義されている変数と参照変数が選択できます。
配列番号	0 ～ 配列数 【0】	選択された変数が配列変数の場合、引数とする配列番号を設定します。

3 **［挿入］をクリックします。**



設定した関数が登録され、左上のコード編集ウィンドウに表示されます。

4 **手順 1 ～3 を繰り返し、演算処理を設定します。**

5 **演算処理の設定が終わったら、［OK］をクリックします。**

テスト計測と調整のポイント（ユニットマクロ）

計測の精度や速度を高めるために、テスト計測して計測結果を確認しながらパラメータを調整します。
詳細結果エリアには、テキストで次の内容が表示されます。

表示項目	説明
判定	判定結果

画像表示エリアに表示する画像は、画像表示設定のサブ画像で指定した画像が表示されます。

サブ画像番号	表示される画像の説明
0～100	任意に定義可能

調整のポイント（ユニットマクロ）

次のポイントを参考にして調整方法を決めてください。

● コンソールウィンドウにエラーメッセージが出る

調整すべきパラメータ	処置
-	「9-1 エラー一覧 (P.9-2)」を参照してください。

●「DEL」、「BS」、「Enter」などをクリックしても動作しない

調整すべきパラメータ	処置
-	コードウィンドウにフォーカスが無い（キー入力カーソルが表示されていない）状態では動作しません。操作したい位置をクリックした上で、ボタンをクリックしてください。

● **文字列に改行コードを含めたい**

調整すべきパラメータ	処置
マクロコード	CR → Chr$(13)LF → Chr$(10) を文字列に加算 (+) してください。

● **位置一覧表示で思うような表示を出せない**

調整すべきパラメータ	処置
マクロコード	位置一覧表示が OFF の場合には、"*MEASUREDISPT" と "*MEASUREDISPG" で記述した表示処理が実行されますが、位置一覧表示が ON の場合は、"*MEASUREDISPG" に記述された内容だけが表示されます。

● **計測中のデータ取得に失敗することがある**

調整すべきパラメータ	処置
マクロコード	SetPollingState 関数を使用して該当する通信処理を止めないと、データを通信処理に横取りされることがあります。 （例）TCP 無手順でデータを受信する場合 以下の順序で処理を記述してください。 ·SetPollingState "TcpNormal", false ·データ受信処理 ·SetPollingState "TcpNormal", true

計測結果の表示や出力内容は、「出力できる計測結果（ユニットマクロ）（P.3-24)」を確認してください。

出力できる計測結果（ユニットマクロ）

ユニットマクロの計測結果を出力する場合は、データ出力や演算の処理項目に、ユニットマクロの演算結果の値を代入して、外部出力してください。

外部参照テーブル（ユニットマクロ）

ユニットマクロの外部参照テーブルには、参照できるデータはありません。

3-1-7　プログラムのセーブとロード

マクロカスタマイズ機能で作成したプログラムは、「ユニットマクロ」処理項目や「ユニット演算マクロ」処理項目ではシーンデータとしてセーブやロードができるほか、［本体に保存］でセンサコントローラに設定を保存することができます。また、一部の機能を除いて、プログラム単体をセーブまたはロードすることができます。プログラム単体をファイルとして取り扱えるため、PC 上でテキストエディタを使用してプログラムの管理や解析ができます。

ユニット演算マクロでのプログラムのセーブとロード

「ユニット演算マクロ」処理項目では、処理ユニットとしてシーンデータにセーブやロードができるほか、処理項目のプロパティ設定画面では、プログラム単体のセーブのみ実行できます。「ユニット演算マクロ」処理項目でセーブしたプログラムは、「ユニットマクロ」処理項目のプロパティ設定画面で読み込むことができます。設定した演算処理は以下の手順で、ファイルに出力します。

1 **ファイルエリアで、[保存] をクリックします。**

2 **ファイルの保存場所とファイル名を指定して、[OK] をクリックします。**

使用上の注意

- 「ユニット演算マクロ」処理項目では、プログラム単体のロードはできません。ロードする場合は、シーンデータをロードしてください。
- 「ユニット演算マクロ」処理項目で保存したプログラムには、作成したプログラムのほかに、処理項目が自動生成したプログラムが含まれています。プログラムの自動生成された部分は変更しないでください。変更した場合は、プログラムを正常にロードできなくなる場合があります。

シーン制御マクロでのプログラムのセーブとロード

シーン制御マクロツールでは、[本体に保存] 実行時にセーブされます。

通信コマンドマクロでのプログラムのセーブとロード

通信コマンドマクロツールでは、[本体に保存] 実行時にセーブできるほか、ツールの設定画面ではプログラム単体のセーブとロードが実行できます。

● プログラムのセーブ

1 **ファイルエリアで、[保存] をクリックします。**

2 **ファイルの保存場所とファイル名を指定して、[OK] をクリックします。**

● プログラムのロード

1 **ファイルエリアで、[読込] をクリックします。**

2 **読み込みたいファイルの保存ファイル（.mcr）を選択し、[OK] をクリックします。**

使用上の注意

通信コマンドマクロツールで保存したプログラムには、作成したプログラムのほかにツールで自動生成されたプログラムが含まれています。プログラムの自動生成された部分は変更しないでください。変更した場合は、プログラムを正常にロードできなくなる場合があります。

ユニットマクロでのプログラムのセーブとロード

「ユニットマクロ」処理項目では、処理ユニットとしてシーンデータにセーブやロードができるほか、処理項目のプロパティ設定画面ではプログラム単体のセーブとロードが実行できます。

● プログラムのセーブ

1 **ファイルエリアで、[保存] をクリックします。**

2 ファイルの保存場所とファイル名を指定して、[OK] をクリックします。

● **プログラムのロード**

1 ファイルエリアで、[読込] をクリックします。

2 読み込みたいファイルの保存ファイル（.mcr）を選択し、[OK] をクリックします。

4

プログラムの基本

4-1　プログラムの基本について 4-2
4-1-1　基本構文 4-2
4-1-2　定数 4-5
4-1-3　変数 4-6
4-1-4　シーン変数、システム変数 4-11
4-1-5　演算子 4-12
4-1-6　式 4-14

4-1 プログラムの基本について

本稿参照の要否	ユニットマクロ	通信コマンドマクロ	シーン制御マクロ	ユニット演算マクロ
	必要			

FH シリーズのマクロカスタマイズ機能では、インタプリタ型のプログラミング言語を使用します。インタプリタ型のプログラミング言語は、コンパイラ型と違いコンパイルするための特別な開発環境がなくてもプログラムを作成できます。C 言語などのプログラミング言語とは違い、FH 上で特別な開発環境がなくてもプログラムを作成して実行できます。
マクロカスタマイズ機能で使用するプログラミング言語は、同じインタプリタ型のプログラミング言語である BASIC がベースになっています。プログラミングの文法は、基本的に BASIC の文法に準拠しており、BASIC プログラミングの文法に独自の文法を加えたものがマクロカスタマイズ機能のプログラミングの文法になります。
以降では、マクロカスタマイズ機能でプログラムを作成するために必要な文法を、基本的なものから順に解説します。

4-1-1 基本構文

プログラムの作成には、最低限知っておく必要がある約束事があります。
以下の基本構文例には、マクロカスタマイズ機能で使用するプログラムの基本的な文法が含まれています。

(基本構文例)

```
*MEASUREDISPT

    Rem テキスト結果表示欄にテキストを表示します
    DrawText "Judgment OK", 1, 0

Return
```

文法の種類	説明
文字	文字は、英文字、数字、記号と特殊記号が使用できます。 詳細は、「文字 (P.4-3)」を参照してください。
行	行番号と文（ステートメント）で構成された単位を行と呼びます。 詳細は、「行 (P.4-3)」を参照してください。
行番号	FH のシステムにロードされるときに、行に対して自動的に付加される番号を行番号と呼びます。 詳細は、「行番号 (P.4-3)」を参照してください。
文（ステートメント）	行に記述されたプログラムの処理の記述を文（ステートメント）と呼びます。 詳細は、「文（ステートメント） (P.4-3)」を参照してください。
ラベル	プログラム中の行に任意の名前を与えて行を特定できるようにしたものをラベルと呼びます。 詳細は、「ラベル (P.4-4)」を参照してください。
サブルーチン	<ラベル>～Return で囲まれた部分をサブルーチンと呼びます。 詳細は、「サブルーチン (P.4-4)」を参照してください。
注釈コメント	Rem 関数またはシングルクォーテーション(') に続く文を注釈コメントと呼びます。 詳細は、「注釈コメント (P.4-4)」を参照してください。

● 文字

文字は、英文字、数字、記号と以下の特殊記号が使用できます。英文字の小文字は、ダブルクォーテーション(") で囲まれた範囲にあるときだけ小文字として区別されます。それ以外の部分では、大文字と小文字の区別はありません。ダブルクォーテーション(") で囲まれた範囲と後述の注釈コメント以外では、必ず半角英数字または半角記号を使用してください。

使用できる特殊記号	説明
コロン(:)	ふたつ以上の文を 1 行に書くときの区切りとして使用します。
コンマ(,)	パラメータが並ぶ場合に、その区切りとして使用します。
セミコロン(;)	出力文中でパラメータが並ぶ場合に、その区切りとして使用します。
アポストロフィ(')	コメント文の先頭に付けて使用します。Rem 文でも同様にコメントを表すことができます。
アスタリスク(*)	ラベル名の先頭に付けて使用します。
スペース( )	マクロ関数とそれに続く引数の間に必ず入れます。その他必要に応じて任意に使用します。ただしマクロ関数名、変数名、数値の途中には入れられません。
ダブルクォーテーション (")	文字列型の値を囲むように前後に付けて使用します。
アンパサンド(&)	整数型変数の型識別子として使用します。整数型の変数名や配列名の末尾に必ず付けてください。 型識別子については、「変数の命名規則 (P.4-7)」の項をご覧ください。
シャープ(#)	実数型の型識別子として使用します。実数型の変数名や配列名の末尾に必ず付けてください。 型識別子については、「変数の命名規則 (P.4-7)」の項をご覧ください。
ドル($)	文字列型変数の型識別子として使用します。文字列型の変数名や配列名の末尾に必ず付けてください。 型識別子については、「変数の命名規則 (P.4-7)」の項をご覧ください。
アットマーク(@)	参照変数の型識別子として使用します。参照変数の変数名の末尾に必ず付けてください。 型識別子については、「変数の命名規則 (P.4-7)」の項をご覧ください。

● 行

行番号と文（ステートメント）で構成された単位を行と呼びます。プログラム中には空行やコメントのみの行があってもプログラムとして問題ありません。1 行に複数の文（マルチステートメント）を記述することもできます。複数の文を記述する場合には、それぞれの文をコロン(:) で区切ってください。

● 行番号

FH のシステムにロードされるときに、行に対して自動的に付加される番号を行番号と呼びます。行番号は、おもにプログラムのデバッグ時にエラー箇所を特定する際に使用します。プログラム作成中は行番号を記載しないでください。

● 文（ステートメント）

行に記述されたプログラムの処理の記述を文（ステートメント）と呼びます。文（ステートメント）は、おもに処理の最小単位である式から構成されます。1 文の長さは 245 文字までです。1 文が 245 文字を超える場合は、エラーとなりプログラムの実行が中止されます。
文には次の 3 種類があります。

文の種類	説明
実行文	マクロ関数を記述し、処理を行う文です。
非実行文	注釈コメントなどを記述し、処理を行わない文です。
ラベル	プログラムの分岐先を定義する文です。

● ラベル

プログラム中の行に任意の名前を与えて行を特定できるようにしたものをラベルと呼びます。基本構文例では、以下の箇所がラベルに該当します。

```
*MEASUREDISPT
```

ラベルを指定することで、Goto <ラベル> マクロ関数を使用して、指定したラベル位置まで処理をジャンプすることができます。

● サブルーチン

<ラベル>～Return で囲まれた部分をサブルーチンと呼びます。サブルーチン内で実際に実行させたい処理を記述します。基本構文例では、以下の箇所がサブルーチンに該当します。

```
*MEASUREDISPT

Return
```

各サブルーチンは、GoSub <ラベル>関数で呼び出すことで実行できます。各単体の処理をサブルーチンとして記述することで、プログラムの視認性がよくなります。
マクロカスタマイズ機能の処理部はサブルーチンで構成されています。このため、GoSub* 関数名と記述することで、該当マクロカスタマイズ機能の処理部分を他のサブルーチンから呼び出して実行できます。
マクロカスタマイズ機能のサブルーチンには、あらかじめシステムで定義されているものがあります。システム定義されたサブルーチンは呼び出されるタイミングが決められています。目的に応じたサブルーチンに処理を記述してください。また、これらのサブルーチンの呼び出されるタイミングの詳細については、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。

● 注釈コメント

Rem 関数またはシングルクォーテーション(') に続く文を注釈コメントと呼びます。プログラム中には、任意にコメント（注釈）行を入れることができます。プログラム中に Rem 関数またはシングルクォーテーション(') を記述すると、その行は注釈コメントとみなされます。
基本構文例では、以下の箇所が注釈コメントに該当します。

```
Rem テキスト結果表示欄にテキストを表示します
```

使用上の注意

- 注釈コメントは、注釈コメント以外のステートメントと 1 行に混在させないでください。注釈コメントが正しく認識されず、プログラムが誤作動を起こす可能性があります。
- プログラム入力エリアでは、注釈コメントが入力できます。
  設定できる文字列は、以下のとおりです。
  「Language setting」で設定した言語と英語が設定可能です。

4-1-2　定数

プログラミングに使用する数値や文字列のうち、変化しない値に固有の名前を与えたものを定数と呼びます。定数は、プログラム中で同じ数値や文字列を何度も使用したいときなどで、かつ数値や文字列が固定値の場合に使用します。

定数の使い方

定数は以下のように使用します。

（使用例）

```
A& = 255
AA& = &h7f

B# = 3.14

C$ = "TEST STRING"
```

定数のデータ型

マクロカスタマイズ機能で使用できる定数は以下のとおりです。

データ型	説明	データ範囲	1 データあたりのバイト数
整数型	符号付き整数型の数値を扱うことができます。	-2147483648 ～ 2147483647	4 バイト
実数型	実数型の数値を扱うことができます。	-1.0E30 ～ 1.0E30	8 バイト
文字列型	文字列を扱うことができます。	255 文字まで	-

整数型定数の表記方法として、10 進数表記を含むいくつかの進数表記があります。マクロカスタマイズ機能で使用できる進数表記は以下のとおりです。

進数	表記方法	表記例	数学上の表記
10 進数	不要	100、3456	100、3456
16 進数	&h	&hff、&h7fff	$(ff)_{16}$、$(7fff)_{16}$
8 進数	&o	&o77、&o3447	$(77)_{8}$、$(3447)_{8}$
2 進数	&b	&b1111、&b01100111	$(1111)_{2}$、$(01100111)_{2}$

参考

定数が 16 進数表記で記述されたプログラムを List 関数で表示させると、16 進数表記部分が 10 進数表記に変換されて表示されます。
（例）「A& = &hff」 の List 関数によるプログラム表示は、「A& = 255」になります。

4-1-3　変数

プログラミングに使用する数値や文字列のうち、変化するデータに固有の名前を与えたものを変数と呼びます。変数は、プログラム中で同じ数値や文字列を何度も使用したいときなどで、かつ数値や文字列の内容が変化する場合に使用します。

変数の使い方

変数は以下のように使用します。

(使用例)

```
Rem 1.変数の宣言
Dim POSITION#(1)

Rem 2.変数への値の代入
A# = 320.0
B# = 310.0

Rem 3.変数の参照
POSITION#(0) = A#
POSITION#(1) = B#
C# = POSITION#(0) + POSITION#(1)
```

(1)変数の宣言

変数を宣言することで、変数に必要なデータ領域を確保して以降のプログラムで使用できるようになります。宣言が必要なのは配列変数の場合だけで、変数の宣言には Dim 関数を使用します。配列変数以外の変数の場合、プログラム実行時に自動で確保されるので宣言は不要です。

また、マクロ変数チェック機能を使って、変数の未定義や重複チェックができます。マクロ変数チェック機能については、「マクロ変数チェック機能 (P.4-9)」の項をご覧ください。

(2)変数への値の代入

変数に値を代入します。代入できる値は、変数のデータ型と同じデータ型の値だけです。配列変数以外の変数の場合、宣言せず変数を使用することができます。名称の末尾に型識別子を加え、他の変数と異なる変数名を使用してください。

型識別子については、「変数の命名規則 (P.4-7)」の項をご覧ください。

(3)変数の参照

変数に代入された値を参照します。マクロ関数へ変数を引数として渡すこともできます。

変数のデータ型

変数を使用するためには、扱うデータを格納するためのデータ領域を変数を使用する前に確保しておく必要があります。確保するデータ領域の大きさとデータをどのように扱うかを定義したものをデータ型と呼びます。

マクロカスタマイズ機能で使用できる変数のデータ型と扱えるデータ範囲は以下のとおりです。データ型は、変数の用途や目的に合わせて選択してください。

データ型	説明	データ範囲	1 データあたりのバイト数
整数型	符号付き整数型の数値を扱うことができます。	-2147483648 ～ 2147483647	4 バイト
実数型	実数型の数値を扱うことができます。	-1.0E30 ～ 1.0E30	8 バイト
文字列型	文字列を扱うことができます。	255 文字まで	-

データ型	説明	データ範囲	1 データあたりのバイト数
バリアント型	不定長のバイナリデータを扱うことができます。	-	-

変数領域サイズ：可変長

使用上の注意

対応バージョン 5.20 より以前のバージョンでは、バリアント型のデータ型は、Syntax error になり使用できません。

変数の命名規則

変数名には命名規則があります。以下の規則にしたがって変数名を決定してください。

変数名		
先頭	途中	末尾
アルファベット ('A'～'Z'、'a'～'z')	アルファベット、数字、記号 ('A'～'Z'、'a'～'z'、'0'～'9'、アンダーライン'_'、ピリオド'.')	型識別子 ('&'、'#'、'$'、'@') *1

*1. '@'は参照変数を表す型識別子であり、プログラム内でユーザが直接宣言した変数に使用することはできません。

変数として扱うデータのデータ型を特定するための識別子を型識別子と呼びます。変数ごとに、変数名の末尾に必ず型識別子を付加してください。使用可能な型識別子については、「変数の種類と型識別子 (P.4-7)」を参照してください。

使用上の注意

- 数字で始まる変数名は使用できません。
- 特殊記号を変数名に使用することはできません。
- 予約語と同じ変数名を使用することはできません。ただし、予約語を含む変数名は使用できます。
  （例）（使用不可）WAIT&→（使用可）WAITTIME&
- 小文字と大文字は、同じ文字として扱われます。
  （例）"AA&" と"aa&" は、同じ変数になります。

変数の種類と型識別子

マクロカスタマイズ機能で使用可能な変数には次のような種類があります。用途や目的に合わせて使用してください。

● 一時変数

ひとつのデータを取り扱うことができ、もっとも基本的な変数です。
一時変数で使用可能な型識別子は以下のとおりです。

データ型	型識別子	表記例
整数型	&	A& = 1
実数型	#	A# = 12.34
文字列型	$	A$ = "OMRON"
バリアント型	!	A! = 1 A! = 12.34 A! = "OMRON"

使用上の注意

型識別子は省略できません。必ず変数名の末尾に型識別子をつけてください。

● 配列変数

複数の同じデータ型のデータをひとまとめにして取り扱いたいとき、一時変数に番号を付与して取り扱うことができます。このような一時変数を配列変数と呼びます。マクロカスタマイズ機能では、最大 4 次元までの配列変数が使用できます。配列の要素番号は 0 から始まり、要素数は「要素番号の数 + 1」になります。また配列変数の要素数は、ReDim 関数を使用することで、変更することもできます。

(例)

```
Rem 1 次元配列、要素は A&(0)、A&(1)、A&(2)、A&(3) の 4 個
Dim A&(3)

Rem 2 次元配列、要素数は 11×11=121
Dim B&(10, 10)

Rem 3 次元配列、要素数は 101×101×101=1030301
Dim C&(100, 100, 100)

Rem 要素数未定義の配列定義
Dim AA&()

Rem 定義済みの配列変数の要素数を変更
Dim AA&()
ReDim AA&(10)
Dim BB&(3)
ReDim BB&(10)
```

配列変数で使用可能な型識別子は以下のとおりです。

データ型	型識別子	表記例
整数型	&()	A&(1) = 1
実数型	#()	A#(2) = 12.34
文字列型	$()	A$(3) = "OMRON"
バリアント型	!()	A!(1) = 1 A!(2) = 12.34 A!(3) = "OMRON"

使用上の注意

- 配列変数名と通常の変数名は、同一名称であっても区別されます。
  (例) 以下の変数は、すべて区別されます。
  A&、A&(10)、A#(10)、A$、A$(10)
- 変数名が同一であれば、異なる次元であっても、同じ配列として扱われます。このとき、最後に定義した配列が有効になります。
  (例) 一番最後に定義された、A&(10,10,10,10) が有効になります。
  A&(10)、A&(10,10)、A&(10,10,10)、A&(10,10,10,10)
- 要素数を未定義で ReDim 関数による再定義を行った場合は、Syntax error になり使用できません。
- 対応バージョン 5.40 より以前のバージョンでは、要素数未定義の配列定義や、ReDim 関数による再定義については、Syntax error になり使用できません。

● 参照変数

参照変数は、処理ユニットデータ、グローバルデータ、システムデータを参照することができるユーザ定義の変数です。（マクロカスタマイズ機能では、処理ユニットデータやシステムデータなどをマクロ関数を使用して、設定したり取得したりすることができます。参照変数は、そのマクロ関数を使用することなく処理ユニットデータやシステムデータを取り扱うことができます。）
参照変数は、ユニットマクロ処理項目、ユニット演算マクロ処理項目またはシーン制御マクロツールのプログラム編集画面であらかじめ設定しておくことが必要です。設定方法は、各処理項目の設定画面を参照してください。
参照変数で使用可能な型識別子は以下のとおりです。

データ型	型識別子	表記例
整数型	@	A@ = 1
実数型	@	A@ = 12.34
文字列型	@	A@ = "OMRON"

使用上の注意

- 通常の変数は型識別子から整数や実数などのデータ型が特定できますが、参照変数は型識別子からデータ型を特定できません。参照変数を使用するときは、割り付けたデータのデータ型を確認してから使用してください。
- 参照変数の設定はシーンデータに保存されますが、参照変数が参照する値そのものは保存されません。値そのものを保存したい場合は、ユーザデータを使用してください。
- 処理ユニットデータやシステムデータで取得のみのデータを参照変数とした場合、その参照変数への代入を行うと Illegal function calll エラーが発生します。
- グローバルデータを整数型または実数型のマクロ変数に代入する場合、グロールデータに対して Val 関数「Val (P.8-321)」を使用してから代入してください。

参考

- 参照変数の型識別子は、「ユニットマクロ」処理項目、または「ユニット演算マクロ」処理項目の処理ユニットのプロパティ設定画面で参照変数名を設定すると、参照変数名に自動的に付加されます。
- 参照変数の設定は、追加できる数に関して制限はありません。ただし、ユニットマクロとユニット演算マクロの処理ユニット 1 個あたり、またはシーン制御マクロの場合、1 シーンあたりの変数名文字数の合計値に上限があります。
  変数名が 10 文字の参照変数の場合、参照変数の設定を 100 個程度追加できます。

マクロ変数チェック機能

マクロ変数チェック機能は、マクロカスタマイズ機能で、変数の未定義や重複をチェックできる機能です。
変数を Dim 変数の書式で定義し、Option Explicit コマンドを実行することで、以降の処理において、変数の未定義や重複のチェックが可能になります。未定義の場合［Undefined variable］のエラー、重複の場合［Duplicated variable］のエラーがシステム状態監視コンソールウィンドウに表示されます。
ユニットマクロ、通信コマンドマクロ、シーン制御マクロで使用できます。
対応バージョンは、5.40 以降になります。
（詳細は、「Dim (P.8-67)」、「Option Explicit (P.8-198)」、「ReDim (P.8-214)」を参照ください。）

● Dim 変数の定義

Option Explicit コマンドを実行すると、定義した変数のみを使用可能とし、定義した変数の重複をチェックします。

そのため、一時変数、配列変数、参照変数を Dim 変数の書式で定義する必要があります。
定義と同時に変数に値を代入することはできません。

(例)

```
Rem マクロ変数のチェック
Option Explicit
Rem Dim 変数による変数の定義
Dim A&
Dim B&
Dim C&
A& = 0
B& = 1
C& = A& + B&
Print Str$(C&)
```

使用上の注意

対応バージョン 5.40 より前のバージョンでは、Dim 変数による変数の定義は Syntax error エラーになり使用できません。ただし、Dim 配列変数については、従来から使用されているのでエラーは発生しません。

● Option Explicit コマンドの実行

Option Explicit コマンドは、コマンドを実行した以降の行でチェック機能が動作します。
コマンド実行の行よりも前で使用している変数は、変数の未定義や重複のチェック機能の範囲外となります。
そのため、コマンド実行の前に使用している変数は、未定義や重複エラーにはなりません。

使用上の注意

- Option Explicit コマンドは、電源投入時やマクロのロード時には無効状態です。マクロが実行され、Option Explicit が実行された後、有効になります。その後、次のマクロがロードあるいは電源 OFF まで有効になります。
- 対応バージョン 5.40 より前のバージョンでは、Option Explicit コマンドの記述は、Syntax error エラーになり使用できません。

4-1-4　シーン変数、システム変数

プログラムの中で、［TDM エディタ］で定義したシーン変数およびシステム変数を使用することができます。
TDM エディタ、シーン変数、システム変数の詳細については、『画像処理システム FH/FHV シリーズ ユーザーズマニュアル（SDNB-719)』の「変数を使用したフロー編集を行う［TDM エディタ]」を参照してください。

シーン変数、システム変数の使い方

シーン変数、システム変数は以下のように使用します。
TDM エディタの変数リストに定義した変数名を使用できます。プログラム中での変数の宣言は不要です。
TDM エディタの変数リストに表示される変数名は、変数の種類と型識別子が付加された状態（変数の種類. 変数名 型識別子）で表示されます。
変数の種類は、SC（シーン変数）、SY（システム変数）の 2 種類です。
型識別子については、『画像処理システム FH/FHV シリーズ ユーザーズマニュアル（SDNB-719)』の「変数を使用する」を参照してください。

(例)

```
SC.AA& = 12
SY.BB& = 34

Print SC.AA&
Print SY.BB&
```

使用上の注意

シーン変数およびシステム変数は、マクロコマンドの実行結果を格納するための引数に直接指定することはできません。
マクロコマンドの実行結果をシーン変数およびシステム変数に格納する場合、以下のように、他の変数に一旦値を格納し、その変数をシーン変数およびシステム変数に格納するようにしてください。

```
GetUnitData 1, 5, TEMP&
SC.AA& = TEMP&
```

4-1-5 演算子

プログラムの中で、演算を表す記号を演算子と呼びます。演算子は変数に処理を加えるときや、変数や定数を計算・比較するときに使用します。

演算子の使い方

演算子は以下のように使用します。
（使用例）

```
A& = 0                          (1)
B& = 2
C& = 4
D& = 35
E& = 0

A& = 1 + 2 + 4 / 2              (2)

F& = (A& + B&) * C&             (5)
                                (3)
If F& < D& Then
   E& = D& AND 31               (4)
Else
   E& = F& AND 31
EndIf
```

(1)代入演算子
代入演算子は、変数に値を代入するときに使用します。
マクロカスタマイズ機能で使用できる代入演算子は、以下のとおりです。

演算子	演算内容	表記例
=	左辺に右辺の値を代入	A& = B& + C&

参考

- If ～Then 文や Select 文などの条件比較で"=" を使用する場合、"=" は左辺と右辺が等しいかどうかを判定する関係演算子として扱われます。
- 整数型変数に実数型の値を代入するときは、小数点以下が四捨五入された値が代入されます。
- 整数型変数に代入される値が-2,147,483,648～2,147,483,647 以外のときは、代入時に Overflow error が発生します
- 実数型に代入される値が-1.0E30 ～1.0E30 以外のときは、代入時に Overflow error が発生します。

(2)算術演算子
算術演算子は、数値データを四則演算、指数演算または剰余演算します。0 による除算を行うとエラーになります。加算、減算、乗算などの算術演算の途中結果が-1.0e30 ～1.0e30 の範囲を超えた場合にはエラーになります。
マクロカスタマイズ機能で使用できる算術演算子は、以下のとおりです。

演算子	演算内容	表記例	数学上の表記
+	加算	A& + B&	A+B
-	減算	A& - B&	A-B
*	乗算	A& * B&	A×B または AB
/	除算	A& / B&	A÷B または A/B
^	指数演算	A& ^ B&	A^B

演算子	演算内容	表記例	数学上の表記
mod	剰余	A& MOD B&	A-[A/B]×B [ ]はガウス記号

(3)関係演算子

関係演算子は、ふたつの数値データまたはふたつの文字データを比較演算します。比較結果が真の場合は(-1)、偽の場合は(0)を返します。通常、If ～Then 文の中で使うことができ、プログラムの流れを制御する場合などに使用できます。

マクロカスタマイズ機能で使用できる関係演算子は、以下のとおりです。

演算子	演算内容	表記例
=	等しい	A& = B&
<>、><	等しくない	A& <> B&、A& >< B&
<	小さい	A& < B&
>	大きい	A& > B&
<=、=<	小さいか等しい	A& <= B&、A& =< B&
>=、=>	大きいか等しい	A& >= B&、A& => B&

参考

If ～Then 文や Select 文などの条件比較のとき以外で"=" を使用する場合、"=" は左辺に右辺の値を代入する代入演算子として扱われます。

(4)論理演算子

論理演算子は、複数の条件を調べるときや、指定数値に対するビット操作や 2 進数演算を行うときに使用します。

マクロカスタマイズ機能で使用できる論理演算子は、以下のとおりです。

演算子	演算内容	表記例
NOT	否定	NOT A&
AND	論理積	A& AND B&
OR	論理和	A& OR B&
XOR	排他的論理和	A& XOR B&

(5)演算子の演算順序

ひとつの式に複数の演算子が含まれる場合、演算は演算子の優先順位にしたがって実行されます。演算の順番を制御したい場合は、先に演算したい内容をカッコで囲むなどしてください。

演算子の優先順位は以下のとおりです。

優先順位	演算子
1	カッコ( )で囲まれたもの
2	マクロ関数
3	指数演算(^)
4	負号(-)
5	乗除算(*、/)
6	剰余算(mod)
7	加減算(+、-)
8	関係演算子(<>、= など)
9	NOT
10	AND
11	OR
12	XOR

4-1-6　式

定数や変数、それらを演算子で結合したものを式と呼びます。結合したものだけでなく定数や変数自身も式であり、式そのものまたは式を組み合わせて文を構成します。

数式

返す値が数値の式を数式と呼びます。数値定数、数値変数、数値を返すマクロ関数を算術演算子、論理演算子で結合して記述します。カッコ( ) で囲むことで、複数の数式を結合することもできます。

(数式の例)

```
A& = 1 + 2 + 4 / 2
```

文字式

返す値が文字列の式を文字式と呼びます。文字列型定数、文字列型変数、文字列を返すマクロ関数をプラス記号で結合して記述します。カッコ( ) で囲むことで、複数の文字式を結合することもできます。

(文字式の例)

```
B$ = "OMRON" + "FH"
```

関係式

ふたつの数式を関係演算子で結合した式を関係式と呼びます。

(関係式の例)

```
If A& < 10 Then

EndIf
```

論理式

複数の関係式を論理演算子で結合した式を論理式と呼びます。ビット操作や 2 進数演算を実行したり、複合的な条件を判定したりする場合に使用します。論理式では式の演算結果が 0 以外の場合は真、0 の場合は偽として扱われます。また、論理式は数値を返すため、数式として扱うこともできます。

(論理式の例)

```
A& = D& AND &b110000
```

関数

あらかじめ定義された命令や演算を実行する式、または指定した特定の値（引数）に対して、あらかじめ定義された演算を実行して演算結果を返す式を一般的に関数と呼びます。特にマクロカスタマイズ機能で使用できる関数をマクロ関数と呼びます。マクロ関数には、戻り値を返さないものと、演算結果を数値または文字列で返すものがあります。
戻り値を返さないものはマクロ関数名と引数をスペースで区切って記述します。戻り値を返すものはマクロ関数名にカッコ( ) で囲んだ引数を続けます。

(戻り値を返さないマクロ関数の例)

```
ChangeScene 1
```

(戻り値を返すマクロ関数の例)

```
C# = Abs(-10)
```

5

マクロプログラム

5-1　マクロプログラムの書き方について 5-2
5-1-1　処理ユニットに関係するデータの種類 5-2
5-1-2　システムに関係するデータの種類 5-7
5-1-3　データのスコープと保存領域 5-9
5-1-4　状態遷移と実行タイミング 5-12
5-1-5　処理の排他制御 5-18

5-1　マクロプログラムの書き方について

マクロカスタマイズ機能は、プログラミングする内容により色々なことに応用できます。基本的なプログラムの書き方に加えて、ここではマクロカスタマイズ機能をより広く使いこなすためのプログラミングの手法について解説します。

5-1-1　処理ユニットに関係するデータの種類

本稿参照の要否	ユニットマクロ	通信コマンドマクロ	シーン制御マクロ	ユニット演算マクロ
	場合によって必要			

マクロカスタマイズ機能のプログラムでは、外部参照データや図形データ、モデルデータなどの処理ユニットのデータを設定または取得することで計測フロー設定を変更することができます。データへのアクセス方法は、データの種類によって異なります。

種類	説明
外部参照データ	処理ユニットの設定値や計測値を設定または取得するためのデータです。 参照：「外部参照データ (P.5-2)」
図形データ	処理ユニットの領域の図形やモデルの図形を設定または取得するためのデータです。 参照：「図形データ (P.5-3)」
モデルデータ	処理ユニットのモデル登録を設定するためのデータです。 参照：「モデルデータ (P.5-5)」
画像データ	処理ユニット自身が保有している画像のデータです。 参照：「画像データ (P.5-6)」

外部参照データ

外部参照データは、処理ユニットの設定値や計測値を設定または取得するためのデータです。外部参照データには処理項目内で重複しない番号が割り当てられているので、処理ユニット番号と外部参照データ番号を指定してデータの設定と取得ができます。
外部参照データの設定と取得は、マクロカスタマイズ機能のプログラムのほか処理ユニットデータ設定処理項目、処理ユニットデータ取得処理項目、通信コマンドからも可能です。
マクロカスタマイズ機能のプログラムから外部参照データを設定または取得するときは、処理ユニットデータ設定マクロ関数または処理ユニットデータ取得マクロ関数を使用します。処理ユニットデータ設定マクロ関数、処理ユニットデータ取得マクロ関数では数値データと文字列データの設定や取得ができます。ただし、計測領域などの図形データ、サーチモデルやエッジモデルなどのモデルデータ、カメラ画像や計測前処理画像などの画像データは、この方法では設定または取得することはできません。図形データ、モデルデータ、画像データの項をご覧ください。

● データの取得

外部参照データの取得方法は以下のとおりです。

（例）ユニット番号 1 番のサーチ処理ユニットの外部参照データ番号 5 番（相関値）を取得する場合

```
Rem 相関値は実数値なので変数の型識別子は#
GetUnitData 1, 5, DATA#
```

（例）ユニット番号 2 番の 2 次元コード処理ユニットの外部参照データ番号 7 番（デコード文字列）を取得する場合

```
Rem デコード文字列は文字列なので変数の型識別子は$
GetUnitData 2, 7, DATA$
```

● データの設定

外部参照データの設定方法は以下のとおりです。

（例）ユニット番号 1 番のサーチ処理ユニットの外部参照データ番号 147 番（サーチ番号）に 1 を設定する場合

```
SetUnitData 1, 147, 1
```

（例）ユニット番号 2 番の 2 次元コード処理ユニットの外部参照データ番号 300 番（インデックス( ) の比較文字列）に「comparison」を設定する場合

```
SetUnitData 2, 300, "comparison"
```

外部参照データには、外部参照データ番号のほかに外部参照データ識別名を使用して設定または取得することもできます。

（例）ユニット番号 1 番のサーチ処理ユニットの外部参照データ識別名「searchNo」（サーチ番号）に 1 を設定する場合

```
SetUnitData, "searchNo", 1
```

外部参照データ番号 147 番に 1 を設定するのと同じ結果になります。

図形データ

図形データは、処理ユニットの領域の図形やモデルの図形を設定または取得するためのデータです。処理項目に複数の図形データがある場合は、図形データに処理項目内で重複しない番号が割り当てられているので、処理ユニット番号と図形番号を指定することで図形データの設定と取得ができます。図形データの設定と取得は、マクロカスタマイズ機能のプログラムのほか処理ユニット図形設定処理項目、処理ユニット図形取得処理項目からも可能です。
マクロカスタマイズ機能のプログラムから図形データを設定または取得するときは、処理ユニット図形設定マクロ関数または処理ユニット図形取得マクロ関数を使用します。
参照：「9-7 図形番号一覧 (P.9-70)」

● 図形データのフォーマット

図形データは配列変数で指定します。配列の各要素の内容は以下のとおりです。

配列要素	説明	
figure(0)	図形データヘッダ情報	図形データのヘッダ情報です。図形の個数と図形データサイズの情報が含まれます。 上位 16 ビット：図形の個数 下位 16 ビット：図形データサイズのバイト数（figure 配列長 × 4） 図形データヘッダ情報 = 図形データサイズのバイト数 + 図形データ個数 × 65536 図形の個数：図形データに含まれる図形の個数を設定します。通常は 1 を設定してください。複数の図形を組み合わせる場合は、2 以上の組み合わせる図形の個数を設定してください。 図形データサイズのバイト数：図形データサイズをバイト数に換算した値を設定します。配列要素数を 4 倍した値を設定してください。 （例）長方形 1 個の場合 配列長 = 5、図形データ個数 = 1 図形データサイズのバイト数 = 5 × 4 = 20 図形データヘッダ情報 = 20 + 1 × 65536 = 65556
figure(1)	図形 0 種別情報	図形 0 データの種別情報です。描画モードと図形種別の情報が含まれます。 上位 16 ビット：描画モード 下位 16 ビット：図形種別 図形種別情報 = 図形種別 + 描画モード × 65536 描画モード：図形の描画が OR モードか NOT モードかを設定します。 通常は 0（OR モード）を設定してください。2 個以上の図形を使用して図形の一部を除外する場合は、2 個目以降の図形で 1（NOT モード）を設定してください。 図形種別：直線や長方形など当該図形の種別を設定します。 （例）長方形 1 個（描画モードが OR）の場合 図形種別 = 8、描画モード = 0 図形種別情報 = 8 + 0 × 65536 = 8
figure(2)	図形 0 データ	図形 0 の図形データです。図形種別によってサイズと内容は異なります。
figure(M)	図形 1 種別情報	図形 1 データの種別情報です。
figure(M+1)	図形 1 データ	図形 1 の図形データです。図形種別によってサイズと内容は異なります。
:	:	:
figure(N*M)	図形 N 種別情報	図形 N データの種別情報です。
figure(N*M+1)	図形 N データ	図形 N の図形データです。図形種別によってサイズと内容は異なります。

● データの取得

図形データの取得方法は以下のとおりです。

（例）ユニット番号 1 番のサーチ処理ユニットの図形データ 1 番（領域の図形）を取得する場合

```
Dim FIGURE&(5)
GetUnitFigure 1, 1, FIGURE&()
```

参考

配列要素数が設定対象または取得対象の図形データに必要な配列要素数に足りない場合は、図形データ設定時または取得時にエラーが発生します。

● データの設定

図形データの設定方法は以下のとおりです。

（例）ユニット番号 1 番のサーチ処理ユニットの図形データ 1 番（領域の図形）の左上座標を（100,50）に変更する場合

```
Dim FIGURE&(10)
GetUnitFigure 1, 1, FIGURE&()

FIGURE&(2) = 100
FIGURE&(3) = 50

SetUnitFigure 1, 1, FIGURE&()
```

（例）ユニット番号 2 番のキズ汚れ処理ユニットの図形データ 0 番（領域の図形（図形 0：長方形、図形 1：楕円））の図形 1 の楕円 X 方向半径を 100 に変更する場合

```
Dim FIGURE&(10)
GetUnitFigure 2, 0, FIGURE&()

Rem FIGURE&(0)：図形データヘッダ情報
Rem FIGURE&(1)：図形 0 の図形種別情報
Rem FIGURE&(2)～FIGURE&(5)：長方形の左上 XY 座標および右下 XY 座標
Rem FIGURE&(6)：図形 1 の図形種別情報
Rem FIGURE&(7)～FIGURE&(8)：楕円の中心点 XY 座標
Rem FIGURE&(9)～FIGURE&(10)：楕円の X 方向半径および Y 方向半径
FIGURE&(9) = 100

SetUnitFigure 2, 0, FIGURE&()
```

モデルデータ

モデルデータは、処理ユニットのモデル登録を設定するためのデータです。処理項目に複数のモデルデータがある場合は、モデルデータに処理項目内で重複しない番号が割り当てられており、処理ユニット番号とモデル番号を指定することでモデルデータを設定できます。
モデルデータの設定は、マクロカスタマイズ機能のプログラムのほか通信コマンドからも可能です。
マクロカスタマイズ機能のプログラムからモデルデータを設定するときは、処理ユニット図形設定マクロ関数またはモデルコピーマクロ関数を使用します。
参照：「9-8 モデル番号一覧 (P.9-72)」

使用上の注意

モデルデータは、モデルの図形やモデルパラメータと密接に関係しています。モデル再登録するときに、モデルの図形やモデルパラメータを事前に設定してください。

● データの取得

モデルデータの取得処理だけを実行することはできません。

● データの設定

データの設定方法は以下のとおりです。

（例）ユニット番号 1 番のサーチ処理ユニットのモデル（モデルの図形：長方形）を現在の計測画像で再登録して、基準位置を(0, 0)、検出点座標を(320, 240) に変更する場合

```
Dim FIGURE&(5)
GetUnitFigure 1, 0, FIGURE&()

Rem モデルの図形を設定することでモデルを再登録
SetUnitFigure 1, 0, FIGURE&()

Rem 基準座標 XY を更新
SetUnitData 1, 129, 0
SetUnitData 1, 130, 0

Rem 検出点座標 XY を更新
SetUnitData 1, 132, 320
SetUnitData 1, 133, 240
```



参考

バリアント型ではモデルデータの取得および変更はできませんが、モデルデータのコピーをすることができます。

（例） 処理ユニット番号 1 番のサーチ処理ユニットのモデルを、ユニット番号 7 番のサーチ処理ユニットのモデルにコピーする場合

```
Rem 変数の定義は不要です。
GetUnitFigure 1, 0, FIGURE!

Rem 別の処理ユニットの図形データに設定します
SetUnitFigure 7, 0, FIGURE!
```

画像データ

画像データは、処理ユニット自身が保有している画像のデータです。
処理項目に複数の画像データがある場合は、画像データに処理項目内で重複しない番号が割り当てられており、処理ユニット番号と画像番号を指定することで画像データを設定できます。
画像データの設定は、マクロカスタマイズ機能のプログラムから可能です。
マクロカスタマイズ機能のプログラムから画像データを設定するときは、画像データの設定マクロ関数を使用します。

● 画像データの種類

画像データには 2 種類あります。

種類	画像番号	説明
計測画像データ	0 ～ 3	計測フローに登録された処理ユニットが、計測実行時に画像処理する対象となる画像データです。通常はカメラ画像入力や計測前処理など「画像入力関連」の処理項目、「画像変換関連」の処理項目が、計測実行時に計測画像 0 に画像を設定し、それ以外の処理項目が設定された画像を使用して画像処理します。 通常は 0 番を使用してください。
処理ユニット画像データ	0 ～ 最大 31	計測画像データとは別に、各処理項目が保有する画像データです。処理項目によって処理ユニット画像データ数と内容は異なります。 2 値化処理した画像やフィルタ画像など画像処理結果を保持したい場合などに使用します。

● データの取得

画像データの取得処理だけを実行することはできません。

● データの設定

画像データの設定方法は以下のとおりです。

（例）ユニット番号 2 番の計測前処理処理ユニットの画像データ 0 番（前処理画像）を計測画像 0 番に設定して以降の処理ユニットが計測画像として使用できるようにする場合

```
SetMeasureImage 0, 2, 0
```

5-1-2　システムに関係するデータの種類

本稿参照の要否	ユニットマクロ	通信コマンドマクロ	シーン制御マクロ	ユニット演算マクロ
	場合によって必要			

マクロカスタマイズ機能のプログラムでは、ユーザデータ、グローバルデータ、システムデータなどシステムが持つデータを設定または取得することで、計測フローに依存しないデータの設定や取得ができます。
データへのアクセス方法は、データの種類によって異なります。

種類	説明
グローバルデータ	マクロ関数を使用して、異なるシーンからデータを設定または取得できるデータです。演算の途中経過などの保存する必要がない一時的な数値や文字列を設定または取得したい場合に使用します。 参照：「グローバルデータ (P.5-7)」
ユーザデータ	ユーザデータ処理項目を使用して、異なるシーンからデータを設定または取得できるデータです。電源断や再起動後も保存しておきたい数値を設定または取得したい場合に使用します。 参照：「ユーザデータ (P.5-8)」
システムデータ	マクロ関数を使用して、異なるシーンからデータを設定または取得できるデータです。電源断や再起動後も保存しておきたい数値や文字列を設定または取得したい場合に使用します。 参照：「システムデータ (P.5-8)」

グローバルデータ

グローバルデータは、マクロ関数を使用して異なるシーンからデータを設定または取得できるデータです。
グローバルデータとして、演算の途中経過などの保存する必要がない一時的な数値や文字列を、設定または取得したい場合に使用します。（詳細は、「5-1-3 データのスコープと保存領域 (P.5-9)」を参照してください。）
異なるシーンから設定または取得したい場合でデータを保存する必要がある場合には、システムデータを使用してください。（詳細は、「システムデータ (P.5-8)」を参照してください。）
グローバルデータを新たに使用する場合は、あらかじめ AddGlobalData 関数で識別名を指定して使用したいグローバルデータを登録してください。グローバルデータの識別名には、半角英数字と半角記号”.”、”_” が使用できます。（詳細は、「AddGlobalData (P.8-23)」を参照してください。）

● データの取得

グローバルデータの取得方法は以下のとおりです。

（例）データ識別名が「GData」の値を取得する場合

```
Rem データ識別名が"GData" のグローバルデータを初期値 10 として登録します
AddGlobalData "GData", 10

Rem データ識別名が"GData" のグローバルデータを取得します
GetGlobalData "GData", GDATA&
```

● データの設定

グローバルデータの設定方法は以下のとおりです。

（例）データ識別名が「GData」の値を設定する場合

```
Rem データ識別名が"GData" のグローバルデータを初期値 10 として登録します
AddGlobalData "GData", 10

Rem データ識別名が"GData" のグローバルデータに 15 を設定します
SetGlobalData "GData", 15
```

ユーザデータ

ユーザデータは、ユーザデータ処理項目を使用して異なるシーンからデータを設定または取得できるデータです。ユーザデータとして、電源断や再起動後も保存しておきたい数値を、設定または取得したい場合に使用します。（詳細は、「5-1-3 データのスコープと保存領域 (P.5-9)」を参照ください。）
異なるシーンから数値を設定または取得したい場合でデータを保存する必要がない場合には、グローバルデータを使用してください。また、数値だけでなく文字列を設定または取得したい場合には、システムデータを使用してください。（詳細は、「グローバルデータ (P.5-7)」、「システムデータ (P.5-8)」を参照ください。）
ユーザデータを使用する前に、ユーザデータツールでユーザデータの初期値を設定しておくことができます。（詳細は、『画像処理システム FH/FHV シリーズ 処理項目機能リファレンスマニュアル (SDNB-713)』の「ユーザデータ」を参照ください。）

● データの取得

ユーザデータは、ユーザデータ処理項目を使用して取得します。（詳細は、『画像処理システム FH/FHV シリーズ 処理項目機能リファレンスマニュアル (SDNB-713)』の「ユーザデータ」を参照ください。）

● データの設定

ユーザデータは、ユーザデータ処理項目を使用して設定します。（詳細は、『画像処理システム FH/FHV シリーズ 処理項目機能リファレンスマニュアル (SDNB-713)』の「ユーザデータ」を参照ください。）

システムデータ

システムデータは、マクロ関数を使用して異なるシーンからデータを設定または取得できるデータです。システムデータとして、電源断や再起動後も保存しておきたい数値や文字列を設定または取得したい場合に使用します。（詳細は、「5-1-3 データのスコープと保存領域 (P.5-9)」を参照してください。）
異なるシーンから設定または取得したい場合でデータを保存する必要がない場合には、グローバルデータを使用してください。また、数値のみを設定または取得したい場合には、ユーザデータを使用してください。（詳細は、「グローバルデータ (P.5-7)」、「ユーザデータ (P.5-8)」を参照してください。）

システムデータを新たに使用する場合は、あらかじめデータ識別名を指定して使用したいシステムデータを登録してください。システムデータのデータ識別名には、データ識別名 0 とデータ識別名 1 があり、それぞれ識別情報 0 と識別情報 1 に対応したデータ識別名をマクロ関数の引数として指定する必要があります。
識別情報とデータ識別名の詳細については、システムデータ一覧をご覧ください。
（詳細は、「第 9 章 マクロリファレンス一覧 (P.9-1)」、「9-3 システムデータ一覧 (P.9-10)」を参照してください。）

● データの取得

システムデータの取得方法は以下のとおりです。

（例）システムデータの「起動シーン番号」の値を取得する場合

```
Rem データ識別名 0 が"Configuration"、データ識別名 1 が"initialSceneNo" のシステムデータを取得します
GetSystemData "Configuration", "initialSceneNo", SDATA&
```

● データの設定

システムデータの設定方法は以下のとおりです。

（例）データ識別名 0 が「PanDA」、データ識別名 1 が「SData」の値を設定する場合

```
Rem データ識別名 0 が"PanDA" のデータとして、データ識別名 1 が"SData" のシステムデータを初期値 0 として登録します
AddSystemData "PanDA", "SData", 0

Rem データ識別名 0 が"PanDA"、データ識別名 1 が"SData" のシステムデータに 5 を設定します
SetSystemData "PanDA", "SData", 5
```

使用上の注意

システムデータに新しくデータを追加する場合は、データ識別名 0 に「PanDA」を指定してください。

5-1-3 データのスコープと保存領域

本稿参照の要否	ユニットマクロ	通信コマンドマクロ	シーン制御マクロ	ユニット演算マクロ
	場合によって必要			

マクロカスタマイズ機能では、目的に応じて多くの種類のデータを使用でき、変数やマクロ関数を使用してデータの設定や取得ができます。
データの設定や取得が実行できる場所やデータが保存される場所は、データの種類によって異なります。
マクロカスタマイズ機能では、プログラムの目的に応じてデータの種類を使い分けることで、変更しやすくメンテナンスしやすいプログラムを作成できます。

データのスコープ

すべてのデータは、どこからでも設定や取得ができるわけではなく、設定や取得を実行できる場所が限定されています。この限定された範囲を「スコープ」と呼び、データごとに定義されたスコープ内でのみ、データの設定と取得ができます。
マクロカスタマイズ機能で使用可能なデータのスコープは以下のとおりです。

スコープ	説明
処理ユニット内 通信コマンドマクロ内	ひとつの処理ユニット内でのみ有効です。 ユニットマクロ処理項目、ユニット演算マクロ処理項目の処理ユニットが計測フローに複数含まれている場合は、それぞれの処理ユニットに存在する同名の変数は、別の変数として扱われます。 通信コマンドマクロツールで複数の通信コマンドマクロが定義されている場合、それぞれの通信コマンドマクロに存在する同名の変数は、同じ変数として扱われます。通信コマンドマクロを実行すると、変数の値は、直前に実行した通信コマンドマクロの実行時に設定した値になります。
シーン内	ひとつのシーン内で有効です。 ユニットマクロ処理項目、ユニット演算マクロ処理項目の処理ユニットが計測フローに複数含まれている場合は、それぞれの処理ユニットから同じデータの設定や取得ができます。 異なるシーンの処理ユニットからは設定や取得はできません。
システム内	システム内で有効です。 ユニットマクロ処理項目、ユニット演算マクロ処理項目の処理ユニットが計測フローに複数含まれている場合は、それぞれの処理ユニットから同じデータの設定や取得ができます。 通信コマンドマクロツールで複数の通信コマンドマクロが定義されている場合は、それぞれの通信コマンドマクロから同じデータの設定や取得ができます。 すべてのシーングループ、シーンから設定や取得ができます。

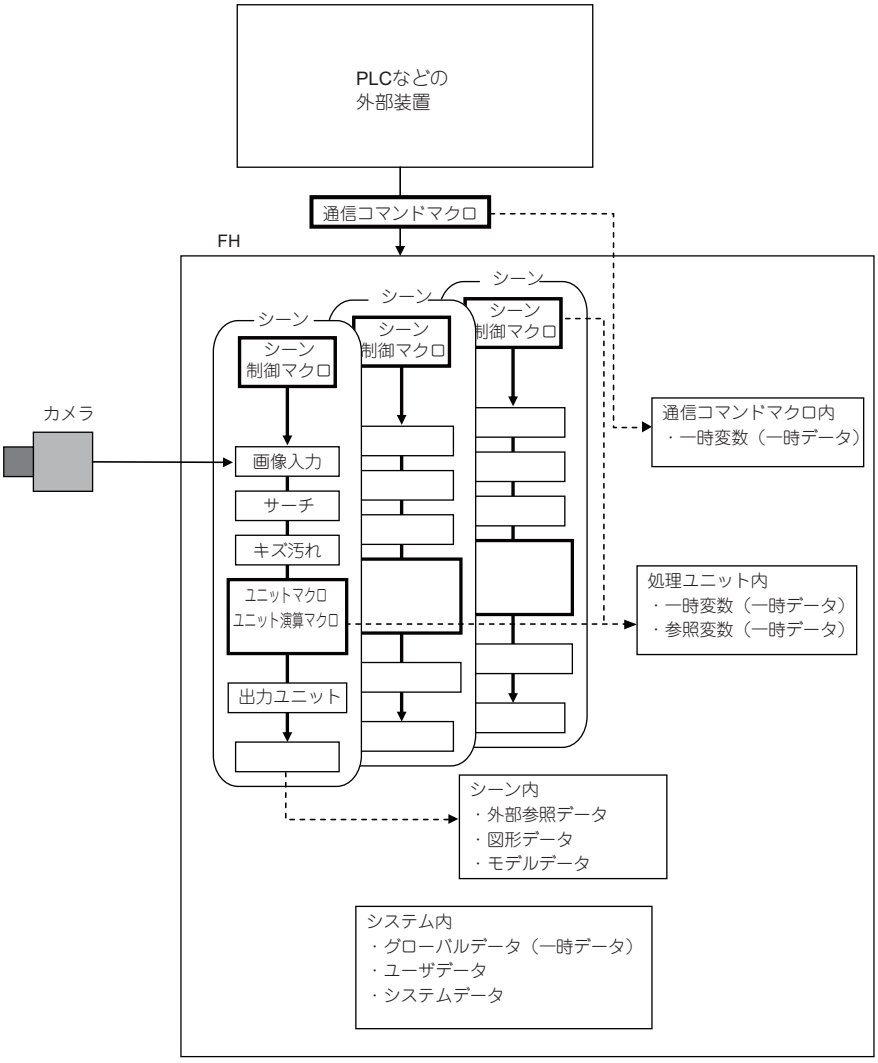
データの保存領域

マクロカスタマイズ機能で使用可能なデータには、メモリ上に一時的にしか保存されないもの、シーンやシステムのデータとして保存されるものがあります。データが保存される場所を「保存領域」と呼び、データごとに保存領域が決められています。
マクロカスタマイズ機能で使用可能なデータの保存領域は以下のとおりです。

保存領域	保存可否	説明
一時データ	保存されない	メインメモリ上に一時的に格納されるデータです。 電源断やシステム再起動でクリアされます。 演算途中の一時的なデータなど、保存する必要のないデータを扱うときに使用してください。
シーンデータ	保存される	シーンデータとして保存されるデータです。 [本体へ保存] で保存することができるだけでなく、シーンデータファイルとしてファイルに保存することができます。 たとえば、計測パラメータなどの品種によって異なる設定値など、シーンごとに保存する必要があるデータを扱うときに使用してください。
システムデータ	保存される	システムデータとして保存されるデータです。 [本体へ保存] で保存することができるだけでなく、システムデータファイルとしてファイルに保存することができます。 たとえば、基準設定などの品種に依存しない検査設定など、シーンごとではなくシーン共通またはシステム共通でデータを扱うときに使用してください。

変数とデータごとのスコープと保存領域

マクロカスタマイズ機能で使用可能な変数とデータごとのスコープと保存領域の関係は以下のとおりです。

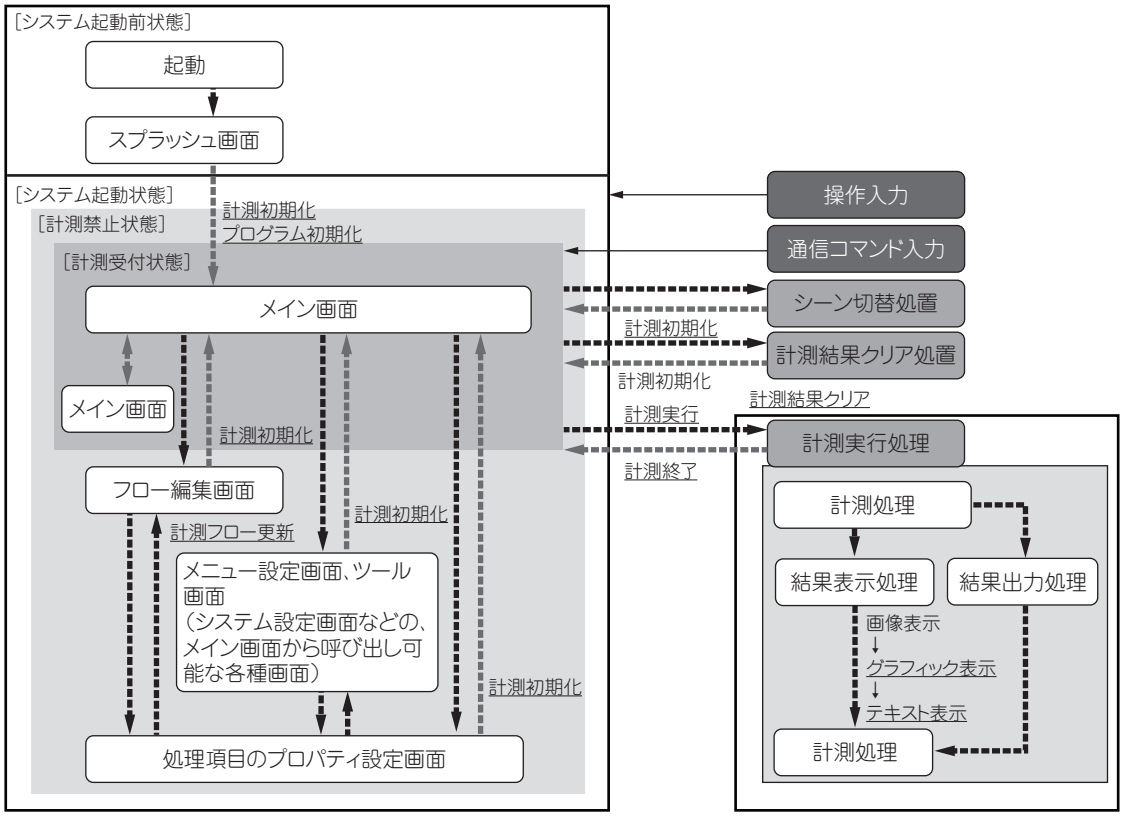


変数とデータの種類	スコープ	保存可否	データ保存領域
一時変数（配列変数） 参照：「一時変数 (P.4-7)」	処理ユニット内 通信コマンドマクロ内	保存されない	一時データ
参照変数 参照：「参照変数 (P.4-9)」			
外部参照データ 参照：「外部参照データ (P.5-2)」	シーン内	保存される	シーンデータ
図形データ 参照：「図形データ (P.5-3)」			
モデルデータ 参照：「モデルデータ (P.5-5)」			
画像データ 参照：「画像データ (P.5-6)」		保存されない	一時データ
グローバルデータ 参照：「グローバルデータ (P.5-7)」	システム内	保存されない	一時データ
ユーザデータ 参照：「ユーザデータ (P.5-8)」		保存される	システムデータ
システムデータ 参照：「システムデータ (P.5-8)」			

5-1-4　状態遷移と実行タイミング

本稿参照の要否	ユニットマクロ	通信コマンドマクロ	シーン制御マクロ	ユニット演算マクロ
	場合によって必要			不要

マクロカスタマイズ機能のプログラムは、いくつかのサブルーチンに分類されます。サブルーチンは、おもに画面や設定、データの状態が遷移するタイミングで実行されます。マクロカスタマイズ機能の各機能により、使用できるサブルーチンが異なります。プログラミングする処理をどのタイミングで実行させたいかに応じて、どのサブルーチンを使用するかを決めてください。



実行タイミングと対応するサブルーチンは以下のとおりです。各実行タイミングの内容は、「サブルーチンの実行タイミングの詳細 (P.5-16)」を参照してください。

サブルーチンの実行タイミング	あらかじめ定義されているサブルーチン名	ユニット演算マクロ	シーン制御マクロ	通信コマンドマクロ	ユニットマクロ	説明
プログラム初期化	*MCRINIT	-	○	-	○	プログラムのロード直後に実行されます。
計測初期化	*MEASUREINIT	-	-	-	○	計測初期化時に実行されます。
計測実行	*MEASUREPROC	○	-	○	○	計測処理実行時に実行されます。
表示						
画像表示	*MEASUREDISPI	-	-	-	○	画像ウィンドウへの画像表示時に呼び出されます。
テキスト表示	*MEASUREDISPT	-	-	-	○	画像ウィンドウへのテキスト表示時に呼び出されます。
グラフィック表示	*MEASUREDISPG	-	-	-	○	画像ウィンドウへのグラフィック表示時に呼び出されます。
計測フロー更新	*RENUMPROC	-	-	-	○	処理ユニット参照番号更新時に呼び出されます。
計測結果クリア	*CLEARMEASUREDATA	-	-	-	○	処理ユニット計測結果初期化時に呼び出されます。

システム状態の遷移と実行の可否

マクロカスタマイズ機能のプログラムには、システムの状態によって実行できる処理と実行できない処理があります。
システムの状態には、以下の種類があります。

状態の種類	説明
計測禁止状態	計測指示や通信コマンドを受け付けることができない状態です。 おもに、起動中や、計測フロー、処理ユニット、システムの設定を変更中または処理や操作を実行中のとき、計測禁止状態になります。 計測禁止状態では、BUSY 信号やそれに類する状態信号は ON になっています。
計測受付状態	計測指示や通信コマンドを受け付けることができる状態です。 計測受付状態では、BUSY 信号やそれに類する状態信号は OFF になっています。

センサコントローラのシステム状態と実行タイミングの関係は以下のとおりです。

サブルーチンの実行タイミング		あらかじめ定義されているサブルーチン名	システムの状態	
			計測禁止状態	計測受付状態
プログラム初期化		*MCRINIT	実行可能	実行不可
計測初期化		*MEASUREINIT	実行可能	実行不可
計測実行		*MEASUREPROC	実行不可	実行可能
表示	画像表示	*MEASUREDISPI	実行可能	実行可能
	テキスト表示	*MEASUREDISPT		
	グラフィック表示	*MEASUREDISPG		
計測フロー更新		*RENUMPROC	実行可能	実行不可
計測結果クリア		*CLEARMEASUREDATA	実行可能	実行不可

サブルーチンの中でも計測禁止状態と計測受付状態を変更することができる場合があります。詳細は、「5-1-5 処理の排他制御 (P.5-18)」を参照してください。

画面の遷移時の実行

マクロカスタマイズ機能のプログラムの実行タイミングには、画面の遷移に関連して発生するものがあります。
画面の遷移には、以下の種類があります。

遷移前の画面	遷移後の画面	説明
スプラッシュ画面	メイン画面	起動時のスプラッシュ画面からメイン画面への画面遷移です。
メイン画面	フロー編集画面 メニュー設定画面 ツール画面 プロパティ設定画面	メイン画面から各画面への画面遷移です。
フロー編集画面	メイン画面	フロー編集画面からメイン画面への画面遷移です。
	プロパティ設定画面	フロー編集画面から処理項目のプロパティ設定画面への画面遷移です。
メニュー設定画面	メイン画面	メニュー設定画面からメイン画面への画面遷移です。
ツール画面	メイン画面	ツール画面からメイン画面への画面遷移です。
	プロパティ設定画面	ツール画面から処理項目のプロパティ設定画面への画面遷移です。

遷移前の画面	遷移後の画面	説明
プロパティ設定画面	メイン画面	処理項目のプロパティ設定画面からメイン画面への画面遷移です。
	フロー編集画面	処理項目のプロパティ設定画面からフロー編集画面への画面遷移です。

センサコントローラの画面遷移とプログラムの実行タイミングの関係は以下のとおりです。

サブルーチンの実行タイミング		あらかじめ定義されているサブルーチン名	画面遷移	
			遷移前の画面	遷移後の画面
プログラム初期化		*MCRINIT	スプラッシュ画面	メイン画面
			プロパティ設定画面（マクロカスタマイズ機能）	フロー編集画面 メイン画面
			ツール画面（マクロカスタマイズ機能）	メイン画面
計測初期化		*MEASUREINIT	スプラッシュ画面 フロー編集画面 メニュー設定画面 ツール画面 プロパティ設定画面	メイン画面
計測実行		*MEASUREPROC	-	-
表示	画像表示	*MEASUREDISPI	-	-
	テキスト表示	*MEASUREDISPT	-	-
	グラフィック表示	*MEASUREDISPG	-	-
計測フロー更新		*RENUMPROC	-	-
計測結果クリア		*CLEARMEASUREDATA	-	-

処理の遷移と実行タイミング

マクロカスタマイズ機能のプログラムの実行タイミングには、画面遷移に関連して発生するものに加えて、実行される処理の種類に関連して発生するものがあります。
処理の種類とプログラムの実行タイミングとの関係は以下のとおりです。

処理	処理時に実行される実行タイミング	処理後に実行される実行タイミング	説明
シーン切替	-	計測初期化	シーン切替処理です。プログラムは、当該処理後の計測初期化のタイミングで実行されます。
計測結果クリア	計測結果クリア	計測初期化	計測結果クリア処理です。プログラムは、当該処理時の計測結果クリアのタイミングと、当該処理後の計測初期化のタイミングでそれぞれ実行されます。
計測実行	計測実行	-	計測実行処理です。プログラムは、当該処理時の計測実行処理のタイミングで実行されます。
	画像表示	-	画像表示処理です。プログラムは、当該処理時の画像表示のタイミングで実行されます。
	グラフィック表示	-	グラフィック表示処理です。プログラムは、当該処理時のグラフィック表示のタイミングで実行されます。
	テキスト表示	-	テキスト表示処理です。プログラムは、当該処理時のテキスト表示のタイミングで実行されます。

処理	処理時に実行される実行タイミング	処理後に実行される実行タイミング	説明
フロー編集	計測フロー更新	-	計測フロー更新処理です。プログラムは、当該処理中のフロー更新のタイミングで実行されます。 計測フローの更新とは、計測フローへの処理ユニットの追加や削除、順番の入れ替えなど計測フローのユニット番号が更新される操作・処理のことです。

メニュー操作と実行タイミング

マクロカスタマイズ機能のプログラムの実行タイミングには、画面の遷移や処理の遷移に関連して発生するものに加えて、実行されるメニュー操作の種類に関連して発生するものがあります。
処理の種類とプログラムの実行タイミングとの関係は以下のとおりです。

処理	処理時に実行される実行タイミング	処理後に実行される実行タイミング	説明
処理ユニット選択	画像表示	-	「フロー表示に連動」が選択している場合の、計測フローでの処理ユニットの選択です。 プログラムは、画像表示のタイミングで実行されます。
	グラフィック表示	-	「フロー表示に連動」が選択している場合の、計測フローでの処理ユニットの選択です。 プログラムは、グラフィック表示のタイミングで実行されます。
	テキスト表示	-	「フロー表示に連動」が選択している場合の、計測フローでの処理ユニットの選択です。 プログラムは、テキスト表示のタイミングで実行されます。

画像表示状態の遷移と実行タイミング

表示に関連したマクロカスタマイズ機能のプログラムの実行タイミングは、画像ウィンドウの表示変更にも関連します。
画像ウィンドウとプログラムの実行タイミングとの関係は以下のとおりです。

処理	処理時に実行される実行タイミング	処理後に実行される実行タイミング	説明
画像モード変更	画像表示 グラフィック表示	計測初期化	画像表示ウィンドウの画像モード変更処理です。プログラムは、当該処理時の画像表示とグラフィック表示のタイミングと、当該処理後の計測初期化のタイミングでそれぞれ実行されます。
画像モード「カメラ画像スルー」	画像表示 グラフィック表示	画像表示 グラフィック表示	画像表示ウィンドウの画像モードが「カメラ画像スルー」のときの処理です。プログラムは、画像モードが「カメラ画像スルー」の間、画像表示とグラフィック表示のタイミングでそれぞれ繰り返し実行されます。

使用上の注意

レイアウト機能で使用可能な 24 個の画像ウィンドウのうち、1 個以上の画像ウィンドウの画像モードが、「カメラ画像スルー」の場合、画像モードの「カメラ画像スルー」の処理が繰り返し実行されます。

サブルーチンの実行タイミングの詳細

あらかじめシステム定義された各サブルーチンの実行タイミングの詳細は以下のとおりです。

● プログラム初期化 (*MCRINIT)

プログラム初期化サブルーチンは、プログラムのロード直後に実行されます。起動時はシーンなどの読み込み途中で実行されるため、処理ユニットやシーン制御マクロにアクセスする処理は正しく動作しません。処理が実行されるのは以下のタイミングです。

サブルーチンの実行タイミングの詳細	おもな実行ケース
マクロ編集の直後	• プログラム編集画面を開いたとき • プログラム編集画面を閉じたとき
シーンデータのロード時	• 起動したとき • シーングループ切替したとき • 以下のファイルのいずれかをロードしたとき シーンデータ シーングループデータ システム + シーングループ 0 データ バックアップデータ

● 計測初期化 (*MEASUREINIT)

計測初期化サブルーチンは、計測初期化時に実行されます。処理が実行されるのは以下のタイミングです。

サブルーチンの実行タイミングの詳細	おもな実行ケース
メイン画面への画面遷移時	• フロー編集画面を閉じたとき • システム設定画面を閉じたとき • その他シーン切替画面や本体に保存画面などのメイン画面から呼び出し可能な各種画面を閉じたとき
レイアウト変更の直後	• レイアウト変更を終了したとき • 画像ウィンドウの設定やテキストウィンドウの設定を変更したとき
設定通信マクロ関数、取得通信マクロ関数の実行時	計測実行、連続実行以外の通信コマンドを実行したとき
計測開始マクロ関数の実行直後	マクロ関数の「MeasureStart」を実行したとき

使用上の注意

画像モードが、「カメラ画像フリーズ」の場合は、計測初期化時に計測画像は更新されません。ただし、レイアウト機能で使用可能な 24 個の画像ウィンドウのうち、ひとつでも「カメラ画像スルー」に設定されている場合は、「カメラ画像フリーズ」の画像ウィンドウの計測画像が更新されます。メイン画面のレイアウト変更機能で、すべての画像ウィンドウの画像モードを「カメラ画像フリーズ」に変更するか、計測初期化処理が実行されるより前のタイミングで、マクロ関数「SetImageWindow」を使用して、すべての画像ウィンドウの画像モードを「カメラ画像フリーズ」に設定してください。
詳細は、『画像処理システム FH/FHV シリーズ ユーザーズマニュアル (SDNB-719)』の「画像モードなどの表示内容を変更する」を参照してください。

● 計測実行 (*MEASUREPROC)

計測実行サブルーチンは、計測処理実行時に実行されます。処理が実行されるのは以下のタイミングです。

サブルーチンの実行タイミングの詳細	おもな実行ケース
計測実行時	• 計測実行、連続計測実行の通信コマンドを実行したとき • メイン画面の計測ボタンやプロパティ設定画面の計測ボタンなどの計測ボタンをクリックしたとき • マクロ関数の「Measure」関数を実行したとき

● 画像表示 (*MEASUREDISPI)

画像表示処理サブルーチンは、画像表示時に実行されます。処理が実行されるのは以下のタイミングです。

サブルーチンの実行タイミングの詳細	おもな実行ケース
画像表示の更新時	• 計測を実行したとき • 画像ウィンドウの設定を変更したとき • 画像ウィンドウを「フロー表示に連動」しているときで、計測フローで選択している処理ユニットを変更したとき

使用上の注意

位置一覧表示が ON のときは、画像表示処理サブルーチンとテキスト表示処理サブルーチンは実行されません。画像表示処理サブルーチンまたはテキスト表示サブルーチンを実行させたい場合は、位置一覧表示を OFF に設定してください。
詳細は、『画像処理システム FH/FHV シリーズ ユーザーズマニュアル (SDNB-719)』の「画像モードなどの表示内容を変更する」を参照してください。

● テキスト表示 (*MEASUREDISPT)

テキスト表示処理サブルーチンは、テキスト表示時に実行されます。処理が実行されるのは以下のタイミングです。

サブルーチンの実行タイミングの詳細	おもな実行ケース
テキスト表示の更新時	• 計測を実行したとき • テキストウィンドウの設定を変更したとき • テキストウィンドウを「フロー表示に連動」しているときで、計測フローで選択している処理ユニットを変更したとき

● グラフィック表示 (*MEASUREDISPG)

グラフィック表示処理サブルーチンは、グラフィック表示時に実行されます。処理が実行されるのは以下のタイミングです。

サブルーチンの実行タイミングの詳細	おもな実行ケース
画面表示の更新時	• 計測を実行したとき • 画像ウィンドウの設定を変更したとき • 画像ウィンドウを「フロー表示に連動」しているときで、計測フローで選択している処理ユニットを変更したとき

● 計測フロー更新 (*RENUMPROC)

計測フロー更新処理サブルーチンは、計測フロー変更時に実行されます。処理が実行されるのは以下のタイミングです。

サブルーチンの実行タイミングの詳細	おもな実行ケース
フロー編集操作の直後	• 計測フローへの処理ユニット追加、削除、移動など、計測フローの処理ユニット順を変更したとき • 「AssignUnit」、「CopyUnit」や「DeleteUnit」など、計測フローの処理ユニット番号や処理ユニットそのものを変更するマクロ関数を実行したとき

● 計測結果クリア (*CLEARMEASUREDATA)

計測結果クリア処理サブルーチンは、計測結果のクリア処理時に実行されます。処理が実行されるのは以下のタイミングです。

サブルーチンの実行タイミングの詳細	おもな実行ケース
計測結果のクリア時	• 計測結果クリア画面で［OK］をクリックしたとき • 計測結果をクリアするマクロ関数を実行したとき

5-1-5 処理の排他制御

本稿参照の要否	ユニットマクロ	通信コマンドマクロ	シーン制御マクロ	ユニット演算マクロ
	不要	場合によって必要		不要

おもに通信コマンドマクロと、シーン制御マクロで使用します。
マクロカスタマイズ機能では、プログラム実行時の処理の中で排他制御することができます。計測処理や通信処理のとき、排他制御を実施しないと、計測処理中に誤った通信コマンドや、画面操作によってシーンが切替えられたり、受信中のデータの内容が次に受信したデータに上書きされたりして、正しい処理ができなくなります。

計測処理の排他制御

プログラム中で処理ユニットデータやシステムデータなどのデータ設定とデータ取得を実行する場合やシーン切替などシーンデータが変更される処理を実行する場合、処理中に計測トリガを受け付けない状態にしてから処理をしてください。もし計測トリガを受け付け可能な状態でデータ設定、データ取得やシーン切替を実行しようとした場合は、設定処理中、取得処理中やシーン切替中に計測トリガが入力されるなどでデータに不整合が発生する、誤った計測結果が出力されるなどの危険があります。計測トリガの入力を受け付けないようにするには MeasureStop 関数を使用し、計測トリガの入力を受け付けるようにするには MeasureStart 関数を使用します。

(例)

```
Rem 計測を停止状態にします
MeasureStop

Rem ここでシーン切替やデータ設定をします

Rem 計測を動作状態にします
MeasureStart
```

使用上の注意

- プログラム中で、シーン切替や処理ユニットデータなどデータの設定または取得を実行するときは、必ず MeasureStop を実行してから、データの設定または取得を実行してください。MeasureStop を実行せずにデータの設定や取得を実行すると、設定処理または取得処理の途中で計測処理が実行される場合があり、設定や取得をしようとするデータに不整合が発生するおそれがあります。
- MeasureStop の後に MeasureStart を実行しなかった場合は、計測トリガを受け付けることができません。BUSY 信号が ON したままの状態になります。
- プログラム中で、MeasureStop の後に計測処理を実行する場合は、MeasureStart を実行してから計測処理を実行してください。MeasureStop の後に MeasureStart を実行せずに計測処理を実行しても、計測処理は実行されません。
- ユニットマクロ、ユニット演算マクロでは、排他制御が不要なプログラムを作成してください。計測実行中に計測禁止状態と計測受付状態を切り替えると、期待した動作をしなくなる可能性があります。

通信処理の排他制御

FH シリーズでは、外部装置との通信処理でポーリング処理を使って状態やデータ送受信を監視しています。
データを送受信するときは、通信処理のポーリング処理を停止状態にしてから処理をしてください。もしポーリング処理を動作させた状態でデータの送受信を実行しようとした場合は、データ受信中に外部機器から別のデータを受信するなどで受信しようとしているデータに不整合が発生する危険があります。
通信処理のポーリング処理の状態を変更するときは、SetPollingState 関数を使用します。

(例) シリアル無手順通信でデータを受信する場合

```
Rem シリアル無手順通信を停止状態にします
SetPollingState "SerialNormal", FALSE

Rem ここでデータを受信します

Rem シリアル無手順通信を動作状態にします
SetPollingState "SerialNormal", TRUE
```

6

デバッグ機能

6-1　デバッグ機能の使い方について 6-2

6-1-1　デバッグの準備 6-2
6-1-2　デバッグの手順 6-4
6-1-3　エラー発生状況の確認 6-4
6-1-4　デバッグの開始 6-5
6-1-5　エラー原因の特定 6-6
6-1-6　エラーの解消 6-9
6-1-7　デバッグの終了 6-9

6-1 デバッグ機能の使い方について

本稿参照の要否	ユニットマクロ	通信コマンドマクロ	シーン制御マクロ	ユニット演算マクロ
	必要			

マクロカスタマイズ機能のプログラムは、不正なマクロ関数の呼び出しやプログラミングミスなどが原因で動作中にエラーになることがあります。エラーが発生したとき、原因を調査してエラーが発生しなくなるようにプログラムを修正することをデバッグと呼びます。マクロカスタマイズ機能にはプログラムのデバッグを支援する機能があり、エラーを効率的に取り除くことができます。

使用上の注意

- システム状態監視コンソールウィンドウの右上にある［×］ボタンをクリックするなどの方法で、システム状態監視コンソールウィンドウを終了しないでください。システムが正常に動作しなくなる可能性があります。もし、システム状態監視コンソールウィンドウを終了してしまった場合は、設定を保存し、センサコントローラを再起動してください。
- システム状態監視コンソールウィンドウにテキストを表示する場合、システム状態監視コンソールウィンドウを最小化していると、処理に時間がかかります。

6-1-1 デバッグの準備

マクロカスタマイズ機能では、プログラムの実行形態を変更し、デバッグに有効なマクロ関数を使用してデバッグを進めます。デバッグには、プログラムにデバッグ用のマクロ関数を記述して実行する方法と、システム状態監視コンソールウィンドウからマクロ関数を直接入力して実行する方法があります。

プログラムの実行形態

マクロカスタマイズ機能では、機能ごとにプログラムの実行形態を指定できます。プログラムを実行する状況によって実行形態を使い分けることで、運用やデバッグがしやすくなります。

実行形態	説明
リリースモード	プログラムの実行や通常の運用の際に使用する実行形態です。 リリースモードでは、プログラムの中に記述されたデバッグ用のマクロ関数は無効化され実行されません。
デバッグモード	プログラムの修正やデバッグを実施する際に便利な実行形態です。 デバッグ機能の中には、デバッグモード時のみ有効になる機能があります。

使用上の注意

プログラムの実行形態は、機能やユニットごとに個別に指定してください。
- ユニット演算マクロ、ユニットマクロの場合：処理ユニットごと
- シーン制御マクロの場合：シーンごと
- 通信コマンドマクロの場合：通信コマンドごと

デバッグ時に使用するマクロ関数

マクロ関数の中にはデバッグに有効な関数があります。プログラムのデバッグに有効なマクロ関数は以下のとおりです。

マクロ関数	説明
Debug（参照：「Debug (P.8-63)」）	プログラムの運用形態とプログラム実行時の情報出力方法を選択します。 • 0：運用形態をリリースモードに設定します。エラー発生時にエラー内容を出力しません。 • 1：運用形態をリリースモードに設定します。エラー発生時にエラー内容をシステム状態監視コンソールウィンドウに出力します。デバッグ時以外は、この設定で使用することをおすすめします。 • 2：運用形態をリリースモードに設定します。プログラムを実行した際に各行の内容をシステム状態監視コンソールウィンドウに出力します。 • 3：運用形態をリリースモードに設定します。エラー発生時にエラー内容をメッセージボックスに出力します。 • 16:運用形態をデバッグモードに設定します。エラー発生時にエラー内容を出力しません。 • 17:運用形態をデバッグモードに設定します。エラー発生時にエラー内容をシステム状態監視コンソールウィンドウに出力します。 • 18:運用形態をデバッグモードに設定します。プログラムを実行した際に各行の内容をシステム状態監視コンソールウィンドウに出力します。通常のデバッグ時には、この設定で使用することをおすすめします。 • 19:運用形態をデバッグモードに設定します。エラー発生時にエラー内容をメッセージボックスに出力します。
Stop（参照：「Stop (P.8-286)」）	このマクロ関数は、デバッグモードでのみ機能が有効になります。 ブレークポイントとしてプログラムの実行を中断します。SetStop 関数であらかじめ条件を設定しておくことで、指定した条件が成り立つ場合だけプログラムの実行を中断することもできます。
SetStop（参照：「SetStop (P.8-270)」）	このマクロ関数は、デバッグモードでのみ機能が有効になります。 Stop 関数でプログラムの実行を中断する条件を設定します。
DebugPrint（参照：「DebugPrint (P.8-64)」）	このマクロ関数は、デバッグモードでのみ機能が有効になります。 システム状態監視コンソールウィンドウにテキストを表示します。

デバッグ時に、システム状態監視コンソールウィンドウでデバッグ用のマクロ関数を入力して実行できます。プログラムを編集することなくシステム状態監視コンソールウィンドウ上でプログラムの実行や中断を指示できるので便利です。
システム状態監視コンソールウィンドウ上で使用すると便利なデバッグ用のマクロ関数は以下のとおりです。

マクロ関数	説明
Cont（参照：「Cont (P.8-49)」）	Stop 関数でプログラムの実行を中断した状態から、プログラムの実行を再開します。 パラメータを指定することで、プログラムをステップ実行できます。 • パラメータなし：プログラムの実行を再開します。 次のプログラムが最後またはエラーが発生するまで実行されます。 • 0：プログラムをステップイン実行します。 現在のプログラム行がサブルーチンの呼び出しの場合は、サブルーチンの中に入ってステップ実行します。それ以外の場合は、現在のステートメントを実行して次のプログラム行で中断します。 • 1：プログラムをステップオーバー実行します。 現在のプログラム行がサブルーチンの呼び出しの場合は、サブルーチン全体を実行してサブルーチン呼び出し後の次のプログラム行で中断します。それ以外の場合は、現在のステートメントを実行して次のプログラム行で中断します。 • 2：プログラムをステップアウト実行します。 現在のプログラム行がサブルーチンから呼び出されたサブルーチンの場合は、現在のプログラム行以降のサブルーチン全体を実行してサブルーチン呼び出し元のサブルーチンの次のプログラム行で中断します。それ以外の場合は、プログラムが最後またはエラーが発生するまで実行されます。
Varlist（参照：「VarList (P.8-322)」）	指定した変数名の変数の情報をシステム状態監視コンソールウィンドウに一覧表示します。

6-1-2　デバッグの手順

プログラムでエラーが発生したとき、以下の流れでプログラムを修正します。

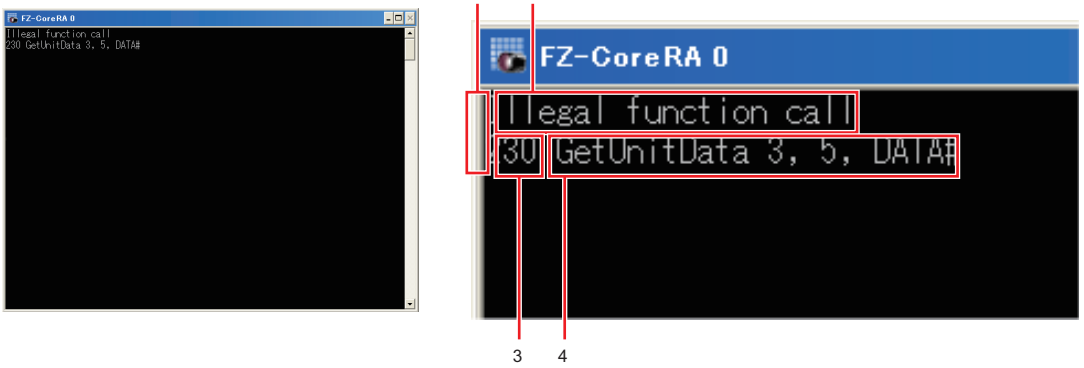
項目	手順	説明
エラー発生	エラー発生	プログラム実行中にエラーが発生し、プログラムの実行処理が強制的に終了します。
	↓	
エラー発生状況の確認（参照：「6-1-3 エラー発生状況の確認 (P.6-4)」）	コンソールウィンドウの確認	発生したエラーの概要がコンソールウィンドウに表示されます。コンソールウィンドウで表示されたエラーの概要を確認します。
	↓	
	エラー内容の特定	エラー内容の特定エラー一覧を参考に、発生したエラーの内容を特定します。
	↓	
デバッグの開始（参照：「6-1-4 デバッグの開始 (P.6-5)」）	デバッグモードに変更	プログラムの*MCRINIT サブルーチンまたは処理の先頭で、Debug 関数をパラメータに 18 を指定して実行します。
	↓	
エラー原因の特定（参照：「6-1-5 エラー原因の特定 (P.6-6)」）	エラー発生箇所の絞り込み	特定したエラー内容をもとに、プログラムのどの行のどのプログラム記述でエラーが発生したかを特定します。
	↓	
	エラー発生原因の特定	エラー発生原因の特定エラーの発生原因を特定します。
	↓	
エラーの解消（参照：「6-1-6 エラーの解消 (P.6-9)」）	修正と動作確認	エラーの原因を解消するためプログラムを修正して、動作確認してエラーが解消していることを確認します。
	↓	
デバッグの終了	リリースモードに変更	プログラムの*MCRINIT サブルーチンまたは処理の先頭で、Debug 関数をパラメータに 1 を指定して実行します。
	↓	
保存	設定を保存	修正した内容を保存します。

6-1-3　エラー発生状況の確認

プログラム実行中にエラーが発生すると、エラーが発生したサブルーチン処理が強制的に終了します。このとき、センサコントローラのエラーが発生した処理以外は、終了せずに処理が続行します。
ユニットマクロ処理項目、またはユニット演算マクロ処理項目でエラーが発生した場合は、エラーが発生した処理ユニットの計測結果が NG になります。通信コマンドマクロでエラーが発生した場合は、マクロ関数がエラーを返します。エラーがどこで発生しても、エラー情報がシステム状態監視コンソールウィンドウに表示されます。

システム状態監視コンソールウィンドウの確認

プログラムでエラーが発生すると、システム状態監視コンソールウィンドウにエラー情報が表示されます。
表示されたエラー情報から、プログラムのどの箇所でどのようなエラーが発生したかを確認することができます。



(1)エラー発生モジュール
エラーを出力するモジュールです。マクロカスタマイズ機能のどの機能がエラーを出力しているかが分かります。

エラー発生モジュールの表示内容	説明
Macro(U**)>	ユニットマクロまたはユニット演算マクロがエラーなどの情報を出力しています。「U」のあとの「**」には処理ユニット番号が表示されます。
Macro(SC)>	シーン制御マクロがエラーなどの情報を出力しています。
Macro(IO)>	通信コマンドマクロがエラーなどの情報を出力しています。

(2)エラーメッセージ
発生したエラーの要因を示すメッセージです。エラーメッセージをもとに、どのようなエラーが発生したかを確認することができます。

(3)エラー行番号
エラーが発生したプログラム行の行番号です。エラーがプログラムのどの場所で発生したかを絞り込むことができます。

(4)エラーステートメント
エラーが発生したプログラム行のプログラム記述内容です。プログラム編集画面を開かなくても、エラーステートメントを確認することでプログラムの記述内容を確認することができます。

エラー内容の絞り込み

システム状態監視コンソールウィンドウに表示されたエラーメッセージをもとに、「9-1 エラー一覧 (P.9-2)」から発生したエラーの内容を確認します。
ここでエラーの原因まで特定できた場合は、エラーの解消を実施します。

使用上の注意

Debug 関数のパラメータで、0 または 16 を指定している場合は、システム状態監視コンソールウィンドウにエラー内容が表示されません。Debug 関数のパラメータで、0 と 16 以外を指定してください。
詳細は、「デバッグ時に使用するマクロ関数 (P.6-2)」を参照してください。

6-1-4 デバッグの開始

エラー発生状況を確認したら、デバッグを開始します。ここでは、ユニットマクロ処理項目の *MEASUREPROC サブルーチンのデバッグを例にしています。
プログラム編集画面を開いて、*MCRINIT サブルーチンかプログラム先頭に Debug 関数を記述してください。

（例）ユニットマクロのデバッグを開始する場合、*MCRINIT サブルーチン内で Debug 関数をパラメータに 18 を指定して記述します。

```
*MCRINIT

    Rem Debug 関数をパラメータに 18 を指定して実行します
    Debug 18

Return
```

（例）通信コマンドマクロのデバッグを開始する場合、デバッグ対象の通信コマンドのプログラム先頭で Debug 関数をパラメータに 18 を指定して記述します。

```
Rem Debug 関数をパラメータに 18 を指定して実行します
Debug 18

Rem ここからは本来の通信コマンドの処理が記述されています
ChangeScene 1
SetUnitData 2, 101, 0
```



参考

- Debug 関数で指定するパラメータには、1 や 18 の他にも種類があります。用途に応じてお使いください。
  詳細は、「デバッグ時に使用するマクロ関数 (P.6-2)」を参照してください。
- プログラムに記述されている Debug 関数を削除した場合は、運用形態は初期値には戻らず、直前に実行された Debug 関数で設定した運用形態のままになります。このような場合は、再起動すると運用形態は初期値に戻ります。

6-1-5　エラー原因の特定

運用形態をデバッグモードに変更したら、次にエラーが発生した原因を特定します。

エラー発生箇所の絞り込み

エラーの原因を特定するために、どこでエラーが発生したか発生箇所を絞り込みます。

1 **プログラム編集画面で、Stop 関数を記述します。**

（例）ユニットマクロの*MEASUREPROC サブルーチン内で計測処理の直前に Stop 関数を記述します。

```
*MEASUERPROC

    Rem Stop 関数を実行してこの行でプログラムの実行を止めます
    Stop

    Rem ここからは本来の計測処理のプログラムが記述されています
    POS.X#=(POS0.X@ + POS1.X@) / 2
    POS.Y#=(POS0.Y@ + POS1.Y@) / 2
    Print POS.Y# / POS.X#

Return
```

2 **[OK] でプログラム編集画面を閉じ、メイン画面に戻ります。**
メイン画面が表示されます。

3 **計測実行します。**
Stop 関数の行を実行すると計測処理が中断します。処理が中断しない場合は、Debug 関数のパラメータでデバッグモードを指定しているかどうかを確認してください。

4 **システム状態監視コンソールウィンドウの表示を確認します。**
システム状態監視コンソールウィンドウの表示は以下のようになります。

```
Macro(U3)210 *MEASUREPROC
Macro(U3)220 Stop
Macro(U3)>
```

5 **プログラムを 1 行ずつステップ実行します。**
システム状態監視コンソールウィンドウに対して、キーボードから「cont 1」と入力します。

```
Macro(U3)210 *MEASUREPROC
Macro(U3)220 Stop
Macro(U3)>cont 1
```

6 **キーボードで［Enter］を押下します。**
プログラムが 1 行ステップ実行され、次の行で中断します。

```
Macro(U3)210 *MEASUREPROC
Macro(U3)220 Stop
Macro(U3)>cont 1
Macro(U3)230 POS.X#=(POS0.X@ + POS1.X@) / 2
Macro(U3)>
```

7 **エラー発生箇所を実行するまでステップ実行を繰り返します。**
エラー箇所までステップ実行すると、システム状態監視コンソールウィンドウにエラー内容が表示されます。ここでは、0 割によるエラーが発生していることが分かります。

```
Macro(U3)210 *MEASUREPROC
Macro(U3)220 Stop
Macro(U3)>cont 1
Macro(U3)230 POS.X#=(POS0.X@ + POS1.X@) / 2
Macro(U3)>cont 1
Macro(U3)240 POS.Y#=(POS0.Y@ + POS1.Y@) / 2
Macro(U3)>cont 1
Macro(U3)250 Print POS.Y# / POS.X#
Macro(U3)Division by zero in 250
Macro(U3)>
```

エラー発生原因の特定

発生箇所を絞り込んだら、次にエラーの原因を特定します。

1 **変数の内容を確認します。**
システム状態監視コンソールウィンドウに対して、キーボードから「VarList」と入力します。

2 **キーボードで［Enter］を押下します。**
システム状態監視コンソールウィンドウに変数の一覧が表示されます。

```
Macro(U3)210 *MEASUREPROC
Macro(U3)220 Stop
Macro(U3)>cont 1
Macro(U3)230 POS.X#=(POS0.X@ + POS1.X@) / 2
Macro(U3)>cont 1
Macro(U3)240 POS.Y#=(POS0.Y@ + POS1.Y@) / 2
Macro(U3)>cont 1
Macro(U3)250 Print POS.Y# / POS.X#
Macro(U3)Division by zero in 250
Macro(U3)>varlist
POS.X#=0.000
POS0.X@=0.000
POS1.X@=0.000
POS.Y#=211.000
POS0.Y@=209.000
POS1.Y@=213.000
```

プログラムの内容と変数の一覧から、分母の変数 POS.X# の値が 0 になっていることが分かります。

特定の条件のときだけプログラムを中断する

プログラムを中断する方法として、特定の条件が成り立ったときだけプログラムを中断させることができます。適切に条件を設定できれば、Stop 関数を使う方法と比べて効率的にエラー発生箇所を絞り込むことができます。

```
*MCRINIT

    Debug 18

    Rem Stop 関数のパラメータが 0 の場合に中断するように、SetStop 関数を使ってプログラムを中断する条件を設定します
    SetStop Str$(0)

Return

*MEASUERPROC

    Rem ここからは本来の計測処理のプログラムが記述されています
    POS.X#=(POS0.X@ + POS1.X@) / 2
    POS.Y#=(POS0.Y@ + POS1.Y@) / 2

    Rem Stop 関数を実行して変数 POS.X# が 0 だった場合にこの行でプログラムの実行を止めます
    Stop Str$(POS.X#)
    Rem Stop 関数を実行して変数 POS.Y# が 0 だった場合にこの行でプログラムの実行を止めます
    Stop Str$(POS.Y#)

    Print POS.Y# / POS.X#

Return
```

参考

Stop 関数で指定できる条件として、関係式の真偽（真：-1、偽：0）を設定することもできます。
（例）変数 CORRELATION& が、60 未満のときにプログラムを中断したい場合：
関係式が、真(-1) のときにプログラムを中断するよう SetStop 関数を設定します。
　SetStop Str$(-1)
変数が、60 未満かどうかを表す関係式 CORRELATION& < 60 が成り立つ場合に、Stop 関数でプログラムを中断します。
　Stop Str$(CORRELATION& < 60)

6-1-6　エラーの解消

エラーが発生した原因が特定できたら、次にエラーを解消するためプログラムを修正します。
プログラムを修正した後は、動作確認を行いエラーが解消されていることを確認してください。

ユニットマクロの*MEASUREPROC サブルーチン内で分母が 0 にならないよう、If 文を使って場合分けします。

```
*MEASUERPROC

   Rem Stop 関数を実行してこの行でプログラムの実行を止めます
   Stop

   Rem ここからは本来の計測処理のプログラムが記述されています
   POS.X#=(POS0.X@ + POS1.X@) / 2
   POS.Y#=(POS0.Y@ + POS1.Y@) / 2

   Rem If 文を使って POS.X# が 0 以外のときだけ割り算を実行するように変更します
   If POS.X# <> 0 Then
      Print POS.Y# / POS.X#
   EndIf

Return
```

6-1-7　デバッグの終了

エラーを解消したら、デバッグを終了します。
プログラム編集画面を開いて、*MCRINIT サブルーチンかプログラム先頭に記述した Debug 関数のパラメータを変更してください。

（例）ユニットマクロのデバッグを終了する場合、*MCRINIT サブルーチン内ですでに記述されている Debug 関数のパラメータを 1 に変更します。

```
*MCRINIT

   Rem Debug 関数をパラメータに 1 を指定して実行します
   Debug 1

Return
```

参考

- Debug 関数で指定するパラメータには、1 や 18 の他にも種類があります。用途に応じてお使いください。
  詳細は、「デバッグ時に使用するマクロ関数 (P.6-2)」を参照してください。
- プログラムに記述されている Debug 関数を削除した場合は、運用形態は初期値には戻らず、直前に実行された Debug 関数で設定した運用形態のままになります。このような場合は、再起動すると運用形態は初期値に戻ります。

7

トラブルシューティング

7-1　トラブルシューティングについて 7-2
7-1-1　プログラミングするときのトラブルシューティング 7-2
7-1-2　動作確認するときのトラブルシューティング 7-3
7-1-3　デバッグするときのトラブルシューティング 7-7
7-1-4　本運用するときのトラブルシューティング 7-8

7-1 トラブルシューティングについて

本稿参照の要否	ユニットマクロ	通信コマンドマクロ	シーン制御マクロ	ユニット演算マクロ
	場合によって必要			

マクロカスタマイズ機能が正しく動作しないときは、次の項目を参考に設定の修正や操作をしてください。

7-1-1 プログラミングするときのトラブルシューティング

マクロカスタマイズ機能で、プログラムを作成する際に起きやすいトラブル現象と対処法の一覧です。

現象	原因	対処
「DEL」、「BS」、「Enter」などをクリックしても動作しない	プログラムウィンドウにフォーカスがない可能性があります。	操作したい位置をクリックしてからボタンをクリックしてください。
USB キーボードを接続して記号が入力できない	USB キーボードのキー配列がキーボード上の表示と異なっている可能性があります。	FH シリーズのキーボードのキー配列は、英語キーボードと同じです。 お使いの USB キーボードのキー配列を英語キーボードのキー配列に読み替えるか、プログラム編集画面で［キーボード］をクリックして表示されるスクリーンキーボードを使用してください。
スクリーンキーボードが表示されるまでに時間がかかる	センサコントローラと外部装置をイーサネットで接続している可能性があります。	センサコントローラを外部装置とイーサネットで接続している場合、スクリーンキーボードの表示に時間がかかる場合があります。 ルータや DNS サーバが存在しないネットワーク環境では、デフォルトゲートウェイと DNS サーバのアドレスに IP アドレスと同じ値を設定してください。
意図しない代入が行われる	同じ名前の変数を多重定義すると、意図しない代入が行われる可能性があります。 たとえば、 *MEASUREPROC と *MEASUREDISPT サブルーチンに、同じ名前の変数を定義してそれぞれの用途で使っている場合、ユーザが意図しない変数代入が行われる可能性があります。	同じ名前の変数を定義しないようにしてください。
変数に値がセットされない	値をセットしようとしている変数名をタイプミスしている可能性があります。	型識別子を含めた変数名が異なると、プログラムの中で別の変数として扱われます。 変数名が間違っていないか確認してください。
Print 関数で出力した文字列や注釈コメントが文字化けする	Print 関数や注釈コメントで日本語と英語以外の文字を使用している可能性があります。	Print 関数では、英語のみ使用可能です。 言語に依存する文字は使用することができません。 （例）ドイツ語の「ä」 注釈コメントでは、表示できる言語は以下のとおりです。 • FH シリーズの場合：［Language setting］で設定した言語と英語が表示可能です。

現象	原因	対処
プログラム編集画面を閉じようとするとエラーが発生して画面が閉じられない	プログラムに全角スペース文字やタブ文字が含まれている可能性があります。	プログラムの中の文字列型の定数や変数以外の場所で全角スペース文字やタブ文字が含まれていると、プログラム編集画面を閉じるときにエラーが発生します。 プログラム編集画面の表示設定で「不可視文字表示」にチェックを入れてください。その後プログラムの中で使用されている全角スペース文字とタブ文字を特定して、プログラムから全角スペース文字とタブ文字を取り除いてください。
	プログラムで使用しているマクロ関数名が間違っている可能性があります。	プログラムの中で使用するマクロ関数名が間違っていると、プログラム編集画面を閉じるときにエラーが発生します。 プログラム編集画面の表示設定で「強調色表示」にチェックを入れてください。その後強調表示されていないマクロ関数名を特定して、正しいマクロ関数名に修正してください。
PC のテキストエディタで作成したプログラムをロードするとプログラムの途中までしかロードできない	作成したプログラムが入力可能文字数を超えている可能性があります。	プログラムをロードするとき、入力可能文字数を超えるプログラムは正しくロードされません。 プログラム編集画面に表示される残入力可能文字数を確認して入力可能文字数を超えないようにプログラムを修正してください。

7-1-2 動作確認するときのトラブルシューティング

マクロカスタマイズ機能で、作成したプログラムを動作確認する際に起きやすいトラブル現象と対処法の一覧です。

ユニット演算マクロを動作確認するときのトラブルシューティング

現象	原因	対処
システム状態監視コンソールウィンドウにエラーメッセージが出る	プログラム実行時にエラーが発生しています。	エラーメッセージをご確認のうえエラーを解消してください。 詳細は、「9-1 エラー一覧 (P.9-2)」を参照してください。 エラーの特定が困難な場合は、デバッグによりエラーの原因を特定してエラーを解消してください。 詳細は、「6-1 デバッグ機能の使い方について (P.6-2)」を参照してください。
ユニット演算マクロ処理ユニットの演算結果が未計測になる	演算子の設定が無効になっている可能性があります。	演算子にチェックを入れて有効にしてください。
	演算の判定条件が正しく設定されていない可能性があります。	演算内容と期待した結果にあった判定条件を設定してください。
システム状態監視コンソールウィンドウを最小化していると計測時間がかかる	システム状態監視コンソールウィンドウを最小化した状態では、システム状態監視コンソールウィンドウにテキストを表示する場合に、ウィンドウを画面上に表示する処理が実行されるため時間がかかります。	システム状態監視コンソールウィンドウを画面上に表示させた状態で計測を実行してください。

シーン制御マクロを動作確認するときのトラブルシューティング

現象	原因	対処
システム状態監視コンソールウィンドウにエラーメッセージが出る	プログラム実行時にエラーが発生しています。	エラーメッセージをご確認のうえエラーを解消してください。 詳細は、「9-1 エラー一覧 (P.9-2)」を参照してください。 エラーの特定が困難な場合は、デバッグによりエラーの原因を特定してエラーを解消してください。 詳細は、「6-1 デバッグ機能の使い方について (P.6-2)」を参照してください。
処理中のデータ受信に失敗することがある	SetPollingState 関数で通信処理を停止させていない可能性があります。	ReceiveData 関数などでデータ取得する場合は、SetPollingState 関数を使用して通信処理を止めてください。 詳細は、「SetPollingState (P.8-260)」を参照してください。
SetUnitFigure 関数などで処理ユニットの図形データを設定できず、登録済みの図形も消去されてしまう	図形を設定するときに計測画像がない可能性があります。	処理ユニットが未計測の状態の場合など、計測画像がないと図形の設定に失敗して以前の設定がクリアされます。 図形を設定する前に、あらかじめ計測を実行するか、ImageUpdate 関数を実行して計測画像がある状態にしてください。 詳細は、「ImageUpdate (P.8-147)」を参照してください。
プログラムの中で計測や再計測の処理が動作しない	プログラムの中で、計測禁止状態にもかかわらず Measure や Remeasure を実行しようとしている可能性があります。	プログラムの中で Measure や Remeasure などを使って計測を実行する場合は、MeasureStop 関数と MeasureStart 関数を使用して計測トリガ入力の禁止状態と許可状態を制御してください。 詳細は、「計測処理の排他制御 (P.5-18)」を参照してください。
システム状態監視コンソールウィンドウを最小化していると計測時間がかかる	システム状態監視コンソールウィンドウを最小化した状態では、システム状態監視コンソールウィンドウにテキストを表示する場合に、ウィンドウを画面上に表示する処理が実行されるため時間がかかります。	システム状態監視コンソールウィンドウを画面上に表示させた状態で計測を実行してください。

通信コマンドマクロを動作確認するときのトラブルシューティング

現象	原因	対処
システム状態監視コンソールウィンドウにエラーメッセージが出る	プログラム実行時にエラーが発生しています。	エラーメッセージをご確認のうえエラーを解消してください。 詳細は、「9-1 エラー一覧 (P.9-2)」を参照してください。 エラーの特定が困難な場合は、デバッグによりエラーの原因を特定してエラーを解消してください。 詳細は、「6-1 デバッグ機能の使い方について (P.6-2)」を参照してください。
処理中のデータ受信に失敗することがある	SetPollingState 関数で通信処理を停止させていない可能性があります。	ReceiveData 関数などでデータ取得する場合は、SetPollingState 関数を使用して通信処理を止めてください。 詳細は、「SetPollingState (P.8-260)」を参照してください。

現象	原因	対処
SetUnitFigure 関数などで処理ユニットの図形データを設定できず、登録済みの図形も消去されてしまう	図形を設定するときに計測画像がない可能性があります。	処理ユニットが未計測の状態の場合など、計測画像がないと図形の設定に失敗して以前の設定がクリアされます。 図形を設定する前に、あらかじめ計測を実行するか、ImageUpdate 関数を実行して計測画像がある状態にしてください。 詳細は、「ImageUpdate (P.8-147)」を参照してください。
通信コマンドマクロが実行されない	コマンドが有効になっていない可能性があります。	通信コマンドマクロツールの設定画面で、実行したいコマンド No. 列にチェックを入れてください。
	シミュレーションソフトを使用している可能性があります。	PC 上では通信機能は動作しません。 通信コマンドマクロで作成したコマンドを実行させる場合は、センサコントローラ上で実行してください。
	プログラムの中で、BUSY ON にチェックが入っているにもかかわらず、Measure や Remeasure を実行しようとしている可能性があります。	コマンドの処理の中で Measure や Remeasure などを使って計測を実行する場合は、あらかじめ通信コマンドマクロツールの設定画面で、実行したいコマンドの BUSY ON チェックを外してください。そのうえで、MeasureStop 関数と MeasureStart 関数を使用して計測トリガ入力の禁止状態と許可状態を制御してください。 詳細は、「計測処理の排他制御 (P.5-18)」を参照してください。
無手順の通信コマンドマクロが実行されない	コマンドとして送信する文字列とコマンド名で設定した文字列が一致していない可能性があります。	コマンドとして、通信コマンドマクロツールの設定画面で設定したコマンド名と同じ文字列を送信してください。
無手順以外の通信コマンドマクロが実行されない	コマンド実行時にコマンドパラメータとして送信する数値とコマンド番号が一致していない可能性があります。	コマンドとして、通信コマンドマクロツールの設定画面に表示されているコマンド番号と同じ数値を 16 進（パラレルの場合は 2 進）で送信してください。
通信コマンドマクロにコマンドパラメータを指定できない	パラレルの通信コマンドマクロにコマンドパラメータを指定しようとしている可能性があります。	パラレルでは通信コマンドマクロにコマンドパラメータを指定できません。以下の方法を検討してください。 • 処理ユニットデータ設定であらかじめ必要なデータを処理ユニットに設定したうえで、コマンドを実行してください。 • パラレル以外の通信手段に変更してください。
通信コマンドマクロを実行しても BUSY ON しない	BUSY ON のチェックが入っていない可能性があります。	通信コマンドマクロツールの設定画面で、実行したいコマンドの BUSY ON にチェックを入れてください。
通信コマンドマクロの設定内容を保存できない	［ファイルに保存］で設定内容を保存しようとしている可能性があります。	通信コマンドマクロの設定内容は、バックアップデータ（.bkd）には保存されません。以下の方法を検討してください。 • ［本体に保存］を実行してください。 • 通信コマンドマクロツールの設定画面で［保存］を実行してください。
標準機能の通信コマンドが実行されない	通信コマンドマクロで作成したコマンド名と標準機能の通信コマンド名が重複している可能性があります。	標準機能の通信コマンド名と通信コマンドマクロ機能で作成したコマンド名が重複している場合は、通信コマンドマクロで作成したコマンドが優先して実行されます。 標準機能の通信コマンドと重複しないコマンド名に変更してください。

現象	原因	対処
システム状態監視コンソールウィンドウを最小化していると計測時間がかかる	システム状態監視コンソールウィンドウを最小化した状態では、システム状態監視コンソールウィンドウにテキストを表示する場合に、ウィンドウを画面上に表示する処理が実行されるため時間がかかります。	システム状態監視コンソールウィンドウを画面上に表示させた状態で計測を実行してください。

ユニットマクロを動作確認するときのトラブルシューティング

現象	原因	対処
システム状態監視コンソールウィンドウにエラーメッセージが出る	プログラム実行時にエラーが発生しています。	エラーメッセージをご確認のうえエラーを解消してください。 詳細は、「9-1 エラー一覧 (P.9-2)」を参照してください。 エラーの特定が困難な場合は、デバッグによりエラーの原因を特定してエラーを解消してください。 詳細は、「6-1 デバッグ機能の使い方について (P.6-2)」を参照してください。
処理中のデータ受信に失敗することがある	SetPollingState 関数で通信処理を停止させていない可能性があります。	ReceiveData 関数などでデータ取得する場合は、SetPollingState 関数を使用して通信処理を止めてください。 詳細は、「SetPollingState (P.8-260)」を参照してください。
SetUnitFigure 関数などで処理ユニットの図形データを設定できず、登録済みの図形も消去されてしまう	図形を設定するときに計測画像がない可能性があります。	処理ユニットが未計測の状態の場合など、計測画像がないと図形の設定に失敗して以前の設定がクリアされます。 図形を設定する前に、あらかじめ計測を実行するか、ImageUpdate 関数を実行して計測画像がある状態にしてください。 詳細は、「ImageUpdate (P.8-147)」を参照してください。
DrawLine 関数などの画像表示ウィンドウ制御のマクロ関数で描画した図形が、実際の座標と異なる位置に描画される	位置ずれ修正などの位置を補正する処理ユニットが計測フローに登録されている可能性があります。	位置ずれ修正などの位置を補正する処理ユニットが計測フローに登録されている場合は、UnitData 関数で取得できる座標値は補正前の座標値になります。 DrawLine 関数などのパラメータに、プログラムを記述するユニットマクロ処理ユニットの処理ユニット番号を指定してください。
表示処理サブルーチンでラベリングの 2 値化画像などの計測結果画像を表示できない	画像データを持たない処理ユニットの計測結果画像を表示させようとしている可能性があります。	ラベリングなどおもな計測処理項目は画像データを持ちません。プログラムの中で画像データを持たない処理項目の計測結果画像を取得することはできません。 計測結果画像を取得して表示するためには、高機能前処理など画像データを持つ処理項目を使用してください。 処理項目ごとに参照できる画像番号と画像の内容は、「9-9 画像番号一覧 (P.9-74)」をご覧ください。
センサコントローラ起動時に「Illegal function call」や「zero devide」などのエラーが発生する	表示処理サブルーチンで初期化されていない値が使用されている可能性があります。	表示処理サブルーチンが未計測の状態で実行される場合、GetGlobalData 関数などのデータ取得処理の失敗や、変数が不定値のまま使われることがあります。 計測後だけ描画処理が実行されるよう表示処理サブルーチンのプログラムを変更してください。

現象	原因	対処
ユニットマクロ処理ユニットのテキスト表示処理と画像表示処理の処理結果が表示されない	位置一覧表示が ON になっている可能性があります。	位置一覧表示が ON の場合は、テキスト表示処理サブルーチンと画像表示処理サブルーチンは実行されません。位置一覧表示を OFF に設定してください。 詳細は、『画像処理システム FH/FHV シリーズ ユーザーズマニュアル (SDNB-719)』の「画像モードなどの表示内容を変更する」を参照してください。
MeasureProc 関数でサーチ処理ユニットの計測を実行すると「Illegal function call」のエラーが発生することがある	サーチ処理で対象物がひとつも見つからなかった可能性があります。	MeasureProc 関数で実行したサーチ処理ユニットの計測処理で検出数が 0 の場合は Illegal function call のエラーが発生します。 プログラムに Try～Catch～End Try 関数を使用して検出数が 0 の場合の処理を作成してください。 詳細は、「Try Catch End Try (P.8-307)」を参照してください。
システム状態監視コンソールウィンドウを最小化していると計測時間がかかる	システム状態監視コンソールウィンドウを最小化した状態では、システム状態監視コンソールウィンドウにテキストを表示する場合に、ウィンドウを画面上に表示する処理が実行されるため時間がかかります。	システム状態監視コンソールウィンドウを画面上に表示させた状態で計測を実行してください。

7-1-3 デバッグするときのトラブルシューティング

マクロカスタマイズ機能で、プログラムをデバッグする際に起きやすいトラブル現象と対処法の一覧です。

現象	原因	対処
システム状態監視コンソールウィンドウにエラーメッセージが出る	プログラム実行時にエラーが発生しています。	エラーメッセージをご確認のうえエラーを解消してください。 詳細は、「9-1 エラー一覧 (P.9-2)」を参照してください。 エラーの特定が困難な場合は、デバッグによりエラーの原因を特定してエラーを解消してください。 詳細は、「6-1 デバッグ機能の使い方について (P.6-2)」を参照してください。
Print 関数で出力した文字列や注釈コメントが文字化けする	Print 関数や注釈コメントで日本語と英語以外の文字を使用している可能性があります。	Print 関数では、英語のみ使用可能です。 言語に依存する文字は使用することができません。 (例) ドイツ語の「ä」 注釈コメントでは、表示できる言語は以下のとおりです。 • FH シリーズの場合：[Language setting] で設定した言語と英語が表示可能です。
Stop 関数でプログラムを中断できない	Debug 関数で指定するモードがリリースモードになっている可能性があります。	Debug 関数で指定できるモードがリリースモードになっている場合は、Stop 関数や DebugPrint 関数の機能が有効になりません。 あらかじめ Debug 関数を実行してリリースモードからデバッグモードに変更してください。 詳細は、「6-1 デバッグ機能の使い方について (P.6-2)」を参照してください。

現象	原因	対処
DebugPrint 関数でデバッグ文字列をシステム状態監視コンソールウィンドウに出力できない	Debug 関数で指定するモードがリリースモードになっている可能性があります。	Debug 関数で指定できるモードがリリースモードになっている場合は、Stop 関数や DebugPrint 関数の機能が有効になりません。 あらかじめ Debug 関数を実行してリリースモードからデバッグモードに変更してください。 詳細は、「6-1 デバッグ機能の使い方について (P.6-2)」を参照してください。
特定のプログラム行が動作しない	ステートメントと注釈コメントを同一行に記述している可能性があります。	ステートメントと注釈コメントを同一行に記述している場合、ステートメントが正しく実行されないことがあります。 注釈コメントは独立した行に記述してください。
	Debug 関数で指定するモードがリリースモードになっている可能性があります。	Debug 関数で指定できるモードがリリースモードになっている場合は、Stop 関数や DebugPrint 関数の機能が有効になりません。 あらかじめ Debug 関数を実行してリリースモードからデバッグモードに変更してください。 詳細は、「6-1 デバッグ機能の使い方について (P.6-2)」を参照してください。
システム状態監視コンソールウィンドウを最小化していると計測時間がかかる	システム状態監視コンソールウィンドウを最小化した状態では、システム状態監視コンソールウィンドウにテキストを表示する場合に、ウィンドウを画面上に表示する処理が実行されるため時間がかかります。	システム状態監視コンソールウィンドウを画面上に表示させた状態で計測を実行してください。

7-1-4 本運用するときのトラブルシューティング

マクロカスタマイズ機能を本運用する際に起きやすいトラブル現象と対処法の一覧です。

現象	原因	対処
システム状態監視コンソールウィンドウにエラーメッセージが出る	プログラム実行時にエラーが発生しています。	エラーメッセージをご確認のうえエラーを解消してください。 詳細は、「9-1 エラー一覧 (P.9-2)」を参照してください。 エラーの特定が困難な場合は、デバッグによりエラーの原因を特定してエラーを解消してください。 詳細は、「6-1 デバッグ機能の使い方について (P.6-2)」を参照してください。
標準機能の通信コマンドが実行されない	通信コマンドマクロで作成したコマンド名と標準機能の通信コマンド名が重複している可能性があります。	標準機能の通信コマンド名と通信コマンドマクロ機能で作成したコマンド名が重複している場合は、通信コマンドマクロで作成したコマンドが優先して実行されます。 標準機能の通信コマンドと重複しないコマンド名に変更してください。
通信コマンドマクロの設定を別のセンサコントローラに反映できない	別のセンサコントローラにバックアップデータ（.bkd）だけをロードしている可能性があります。	通信コマンドマクロの設定内容は、バックアップデータ（.bkd）には保存されません。 通信コマンドマクロツールの設定画面で［保存］を実行してセーブした設定ファイル（.mcr）をバックアップデータ（.bkd）とは別にセンサコントローラにロードしてください。 設定ファイル（.mcr）のロードには、設定をロードしたいセンサコントローラの通信コマンドマクロツールの設定画面で［読込］を実行してください。

現象	原因	対処
通信コマンドマクロの設定内容を保存できない	［ファイルに保存］で設定内容を保存しようとしている可能性があります。	通信コマンドマクロの設定内容は、バックアップデータ（.bkd）には保存されません。以下の方法を検討してください。 • ［本体に保存］を実行してください。 • 通信コマンドマクロツールの設定画面で［保存］を実行してください。
保存したはずの計測フローの設定が再起動後クリアされる	シーン制御マクロのプログラムの中に*SaveProcサブルーチンが記述されている可能性があります。	*SaveProc サブルーチンが記述されていると、［本体に保存］やシーンデータの保存のときに*SaveProc サブルーチンで記述した処理だけが実行されます。そのため通常の［本体に保存］や［ファイルに保存］のようにシーンデータ全体の変更が保存されなくなります。 *SaveProc サブルーチンを削除してください。
MeasureProc 関数でサーチ処理ユニットの計測を実行すると「Illegal function call」のエラーが発生することがある	サーチ処理で対象物がひとつも見つからなかった可能性があります。	MeasureProc 関数で実行したサーチ処理ユニットの計測処理で検出数が 0 の場合は Illegal function call のエラーが発生します。 プログラムに Try～Catch～End Try 関数を使用して検出数が 0 の場合の処理を作成してください。 詳細は、「Try Catch End Try (P.8-307)」を参照してください。
システム状態監視コンソールウィンドウを最小化していると計測時間がかかる	システム状態監視コンソールウィンドウを最小化した状態では、システム状態監視コンソールウィンドウにテキストを表示する場合に、ウィンドウを画面上に表示する処理が実行されるため時間がかかります。	システム状態監視コンソールウィンドウを画面上に表示させた状態で計測を実行してください。

8

マクロ関数

8-1　マクロ関数一覧 8-2
8-1-1　アルファベット順 8-2
8-1-2　インデックス（機能別） 8-13
8-2　コマンド詳細 8-21

8-1　マクロ関数一覧

マクロカスタマイズ機能で使用できるマクロ関数の一覧です。

8-1-1　アルファベット順

関数・コマンド名	機能	種別	ユニット演算マクロ	シーン制御マクロ	通信コマンドマクロ	ユニットマクロ	参照先
Abs	指定した数式の絶対値を求めます。	数値演算	○	○	○	○	P. 8-21
AbsolutePath$	クイックアクセスの識別名に割り当てられた絶対パスを取得します。	その他	-	○	○	○	P. 8-22
AddGlobalData	グローバルデータを追加します。	グローバルデータ	○	○	○	○	P. 8-23
AddSystemData	システムデータを追加します。	システムデータ	○	○	○	○	P. 8-24
AND	指定したふたつの式の論理積を求めます。	数値演算	○	○	○	○	P. 8-25
ApplicationPath$	アプリケーションソフトに関連したパス名を取得します。	その他	○	○	○	○	P. 8-26
ApplicationVersion$	アプリケーションソフトのバージョン情報を取得します。	その他	○	○	○	○	P. 8-27
ApproximationCircle	近似円を求めます。	数値演算	○	○	○	○	P. 8-28
ArrayDims	配列の次元数を取得します。	その他	-	○	○	○	P. 8-29
ArrayLen	配列の要素数を取得します。	その他	-	○	○	○	P. 8-30
Asc	指定した文字のキャラクタコードの値を求めます。	文字列演算	○	○	○	○	P. 8-31
AssignUnit	処理ユニットを登録します。	処理フロー制御	-	○	○	-	P. 8-32
Atn	指定した数式の逆正接（アークタンジェント）を求めます。	数値演算	○	○	○	○	P. 8-33
BusyOut	処理実行中信号の出力状態を設定します。	入出力制御	-	○	○	○	P. 8-34
Call（構造化サブルーチン呼出）	指定したサブルーチンに引数を付加して実行します。	一般命令	○	○	○	○	P. 8-35
Call（ユーザ定義関数実行）	登録されているユーザ定義関数を実行します。	デバッグ	○	○	○	○	P. 8-36
ChangeScene	シーンを切り替えます。	シーン制御	-	-	○	-	P. 8-38
ChangeSceneGroup	シーングループを切り替えます。	シーングループ制御	-	-	○	-	P. 8-39
CheckUnit	処理ユニットの登録状態をチェックします。	処理フロー制御	-	○	○	-	P. 8-40
Chr$	指定したキャラクタコードに対応する文字を求めます。	文字列演算	○	○	○	○	P. 8-41
ClearMeasureData	処理ユニットの計測結果をクリアします。	計測制御	-	○	○	○	P. 8-42

関数・コマンド名	機能	種別	ユニット演算マクロ	シーン制御マクロ	通信コマンドマクロ	ユニットマクロ	参照先
ClearScene	シーンデータをクリアします。	シーン制御	-	-	○	-	P. 8-43
ClearSceneGroup	シーングループデータをクリアします。	シーングループ制御	-	-	○	-	P. 8-44
Close	ファイルを閉じます。	ファイル制御	○	○	○	○	P. 8-45
CloseTextData	メッセージファイルを閉じます。	多国語対応メッセージ	○	-	-	○	P. 8-46
Cls	コンソールウィンドウの表示をクリアします。	その他	○	○	○	○	P. 8-48
Cont	プログラムの実行を中断した状態からプログラムの実行を再開します。	デバッグ	○	○	○	○	P. 8-49
ContinuousMeasure	連続計測の開始／停止を行います。	計測制御	-	○	○	-	P. 8-50
CopyMeasureImage	計測画像をユニットマクロ処理ユニットの画像としてコピーします。	処理ユニット制御	-	-	-	○	P. 8-51
CopyScene	シーンデータをコピーします。	シーン制御	-	-	○	-	P. 8-52
CopySceneGroup	シーングループデータをコピーします。	シーングループ制御	-	-	○	-	P. 8-53
CopyUnit	処理ユニットをコピーします。	処理フロー制御	-	○	○	-	P. 8-54
CopyUnitFigure	処理ユニットの図形データをコピーします。	処理ユニット制御	-	○	○	○	P. 8-55
CopyUnitImage	処理ユニットの画像をユニットマクロ処理ユニットの画像としてコピーします。	処理ユニット制御	-	-	-	○	P. 8-56
CopyUnitModel	処理ユニットのモデルデータをコピーします。	処理ユニット制御	-	○	○	○	P. 8-57
Cos	指定した数式の余弦（コサイン）を求めます。	数値演算	○	○	○	○	P. 8-59
Crspoint	2 直線の交点を求めます。	数値演算	○	○	○	○	P. 8-60
Date$	内蔵クロックの日付を読み出します。	数値演算	○	○	○	○	P. 8-62
Debug	プログラムの実行形態と情報出力方法を設定します。	デバッグ	○	○	○	○	P. 8-63
DebugPrint	システム状態監視コンソールウィンドウにデバッグ情報を出力します。	デバッグ	○	○	○	○	P. 8-64
DeleteProfileData	設定ファイル（.ini ファイル）のデータを削除します。	その他	-	○	○	○	P. 8-65
DeleteUnit	処理ユニットを削除します。	処理フロー制御	-	○	○	-	P. 8-66
Dim	配列変数を定義します。	一般命令	○	○	○	○	P. 8-67
DisableUnit	指定ユニットの計測 ON/OFF を切り替えます。	処理フロー制御	-	○	○	-	P. 8-68
DisplaySubNo	画像表示サブ番号を取得します。	画像表示ウィンドウ制御	-	-	-	○	P. 8-69

関数・コマンド名	機能	種別	ユニット演算マクロ	シーン制御マクロ	通信コマンドマクロ	ユニットマクロ	参照先
DisplayUnitNo	フロー表示ウィンドウの処理ユニット番号の選択状態を取得します。	表示制御	-	○	○	-	P. 8-70
Do Loop While	条件を満たしている間、Do から Loop までの間の文を繰り返し実行します。	一般命令	○	○	○	○	P. 8-71
Dposline	指定した直線と点の最短距離を求めます。	数値演算	○	○	○	○	P. 8-72
DrawArc	画像表示ウィンドウに円弧を描画します。	画像表示ウィンドウ制御	-	-	-	○	P. 8-74
DrawArcW	画像表示ウィンドウに幅付き円弧を描画します。	画像表示ウィンドウ制御	-	-	-	○	P. 8-75
DrawBox	画像表示ウィンドウに長方形を描画します。	画像表示ウィンドウ制御	-	-	-	○	P. 8-77
DrawCircle	画像表示ウィンドウに円形を描画します。	画像表示ウィンドウ制御	-	-	-	○	P. 8-79
DrawCircleW	画像表示ウィンドウに幅付き円形を描画します。	画像表示ウィンドウ制御	-	-	-	○	P. 8-80
DrawCursor	画像表示ウィンドウに十字カーソルを描画します。	画像表示ウィンドウ制御	-	-	-	○	P. 8-82
DrawEllipse	画像表示ウィンドウに楕円形を描画します。	画像表示ウィンドウ制御	-	-	-	○	P. 8-83
DrawFigure	画像表示ウィンドウに図形を描画します。	画像表示ウィンドウ制御	-	-	-	○	P. 8-85
DrawFillImage	画像表示ウィンドウに塗り潰し画像を描画します。	画像表示ウィンドウ制御	-	-	-	○	P. 8-86
DrawJudgeText	テキスト表示ウィンドウに判定結果の文字列を描画します。	テキスト表示ウィンドウ制御	-	-	-	○	P. 8-87
DrawLine	画像表示ウィンドウに直線を描画します。	画像表示ウィンドウ制御	-	-	-	○	P. 8-88
DrawLineW	画像表示ウィンドウに幅付き直線を描画します。	画像表示ウィンドウ制御	-	-	-	○	P. 8-90
DrawMeasureImage	画像表示ウィンドウに計測画像を描画します。	画像表示ウィンドウ制御	-	-	-	○	P. 8-91
DrawPoint	画像表示ウィンドウに点を描画します。	画像表示ウィンドウ制御	-	-	-	○	P. 8-92
DrawPolygon	画像表示ウィンドウに多角形を描画します。	画像表示ウィンドウ制御	-	-	-	○	P. 8-93
DrawSearchFigure	画像表示ウィンドウにサーチ図形を描画します。	画像表示ウィンドウ制御	-	-	-	○	P. 8-95
DrawText	テキスト表示ウィンドウに文字列を描画します。	テキスト表示ウィンドウ制御	-	-	-	○	P. 8-97
DrawTextG	画像表示ウィンドウに文字列を描画します。	画像表示ウィンドウ制御	-	-	-	○	P. 8-99
DrawUnitImage	画像表示ウィンドウに処理ユニット画像を描画します。	画像表示ウィンドウ制御	-	-	-	○	P. 8-100
Dskf	ドライブの空き容量を求めます。	ファイル制御	○	○	○	○	P. 8-102
ElapsedTime	計測開始からの経過時間を取得します。	その他	○	-	-	○	P. 8-103

関数・コマンド名	機能	種別	ユニット演算マクロ	シーン制御マクロ	通信コマンドマクロ	ユニットマクロ	参照先
Eof	ファイルの終わりを調べます。	ファイル制御	○	○	○	○	P. 8-104
Erase	配列変数を解放します。	一般命令	○	○	○	○	P. 8-105
Errcmnd$	エラーが発生したマクロ関数名を取得します。	一般命令	○	○	○	○	P. 8-106
Errno	エラー番号を取得します。	一般命令	○	○	○	○	P. 8-107
ErrorOut	ERROR 信号の出力状態を設定します。	入出力制御	○	○	○	○	P. 8-108
ExecuteErrorProc	指定した種別のエラーが発生したことをシステムに通知します。	その他	-	○	○	-	P. 8-110
ExecuteImageLogging	画像ロギングを実行します。	その他	-	-	-	○	P. 8-112
ExitFzProcess	センサコントローラを終了させます。	その他	-	○	○	-	P. 8-113
Exp	自然対数の底 e を底とする指数関数の値を求めます。	数値演算	○	○	○	○	P. 8-114
Fcopy	ファイルをコピーします。	ファイル制御	○	○	○	○	P. 8-115
FigureType	処理ユニットに設定可能な図形種類を取得します。	その他	○	-	-	○	P. 8-116
Fix	数値の小数点以下を切り捨てた整数値を求めます。	数値演算	○	○	○	○	P. 8-117
For To Step Next	For 文と Next 文の間の文を繰り返し実行します。	一般命令	○	○	○	○	P. 8-118
GetAll	すべての入力端子の入力状態を取得します。	入出力制御	-	○	○	○	P. 8-120
GetGlobalData	グローバルデータを取得します。	グローバルデータ	○	○	○	○	P. 8-121
GetImageSize	処理ユニットの画像の画像サイズを取得します。	処理ユニット制御	○	○	○	○	P. 8-122
GetImageWindow	画像表示ウィンドウの状態を取得します。	表示制御	-	○	○	○	P. 8-123
GetMeasureOut	計測結果の外部出力設定を取得します。	計測制御	-	○	○	-	P. 8-125
GetPlcData	ReadPlcMemory 関数で読み込んだデータを取り出します。	入出力制御	-	○	○	○	P. 8-126
GetPollingState	通信モジュールの実行状態を取得します。	入出力制御	-	○	○	○	P. 8-127
GetPort	指定した入力端子の入力状態を取得します。	入出力制御	-	○	○	○	P. 8-129
GetProfileData	設定ファイル（.ini ファイル）のデータを取得します。	その他	-	○	○	○	P. 8-130
GetSceneData	シーン制御マクロに関するデータやシーン変数の値を取得します。	シーン制御	○	○	○	○	P. 8-131
GetSceneGroupData	指定した識別名のシーングループデータの内容を取得します。	シーングループ制御	-	○	○	-	P. 8-132
GetSystemData	システムデータまたは、システム変数の値を取得します。	システムデータ	○	○	○	○	P. 8-133
GetText$	メッセージファイルからテキストデータを取得します。	多国語対応メッセージ	○	-	-	○	P. 8-134

関数・コマンド名	機能	種別	ユニット演算マクロ	シーン制御マクロ	通信コマンドマクロ	ユニットマクロ	参照先
GetTextWindow	テキスト表示ウィンドウの状態を取得します。	表示制御	-	○	○	○	P. 8-137
GetUnitData	処理ユニットのデータを取得します。	処理ユニット制御	○	○	○	○	P. 8-138
GetUnitFigure	処理ユニットの図形データを取得します。	処理ユニット制御	○	○	○	○	P. 8-139
Gosub	指定したサブルーチンを実行します。	一般命令	○	○	○	○	P. 8-140
Goto	指定したラベルの行へ処理を移します。	一般命令	○	○	○	○	P. 8-141
Hex$	数式の値を 16 進数表現の文字列に変換します。	文字列演算	○	○	○	○	P. 8-142
If Then Else	指定した条件によって処理の流れを制御します。	一般命令	○	○	○	○	P. 8-143
If Then Elseif Else Endif	指定した条件によって処理の流れを制御します。	一般命令	○	○	○	○	P. 8-144
ImageFormat	処理ユニットの画像の画像フォーマットを取得します。	処理ユニット制御	○	○	○	○	P. 8-146
ImageUpdate	カメラからの入力画像を更新します。	計測制御	-	○	○	-	P. 8-147
Input#	ファイルからデータを読み込みます。	ファイル制御	○	○	○	○	P. 8-149
Input$	ファイルからバイナリデータを読み込みます。	ファイル制御	○	○	○	○	P. 8-150
InsertUnit	処理ユニットを挿入します。	処理フロー制御	-	○	○	-	P. 8-152
Int	数値を整数値に変換します。	数値演算	○	○	○	○	P. 8-153
Isfile	ファイルの属性と有無を調べます。	ファイル制御	○	○	○	○	P. 8-154
ItemCount	使用可能な処理項目の種類の数を取得します。	処理項目制御	-	○	○	-	P. 8-155
ItemIdent$	処理項目の識別名を取得します。	処理項目制御	-	○	○	-	P. 8-156
ItemInfo	処理項目の情報を取得します。	処理項目制御	-	○	○	-	P. 8-157
ItemTitle$	処理項目のタイトル名を取得します。	処理項目制御	-	○	○	-	P. 8-158
JudgeOut	総合判定信号の出力状態を設定します。	入出力制御	-	○	○	○	P. 8-159
Kill	ファイルまたはフォルダを削除します。	ファイル制御	○	○	○	○	P. 8-160
LanguageString$	現在の言語設定にしたがった言語で文字列を取得します。	その他	○	○	○	○	P. 8-161
LCase$	大文字を小文字に変換します。	文字列演算	○	○	○	○	P. 8-162
Left$	文字列の左側から指定した長さの文字列を取り出します。	文字列演算	○	○	○	○	P. 8-163
Len	指定した文字列の長さを求めます。	文字列演算	○	○	○	○	P. 8-164
LineInput#	ファイルから 1 行分のデータを読み込みます。	ファイル制御	○	○	○	○	P. 8-165

関数・コマンド名	機能	種別	ユニット演算マクロ	シーン制御マクロ	通信コマンドマクロ	ユニットマクロ	参照先
List	システム状態監視コンソールウィンドウにプログラム内容の全部または一部を出力します。	デバッグ	○	○	○	○	P. 8-167
LoadBackupData	システム+シーングループ 0 データを読み込みます。	データセーブ／ロード	-	-	○	-	P. 8-168
LoadScene	シーンデータを読み込みます。	データセーブ／ロード	-	-	○	-	P. 8-169
LoadSceneGroup	シーングループデータを読み込みます。	データセーブ／ロード	-	-	○	-	P. 8-170
LoadSystemData	システムデータを読み込みます。	データセーブ／ロード	-	○	○	-	P. 8-171
LoadUnitData	処理ユニットデータを読み込みます。	データセーブ／ロード	-	○	○	-	P. 8-172
Log	自然対数値を求めます。	数値演算	○	○	○	○	P. 8-174
Lsqumeth	複数点の座標から最小二乗法を使って近似直線を求めます。	数値演算	○	○	○	○	P. 8-175
Measure	計測処理を実行します。	計測制御	-	-	○	-	P. 8-176
MeasureDispG	処理ユニットの計測結果の表示処理を実行します。	処理ユニット制御	-	-	-	○	P. 8-177
MeasureDispI	処理ユニットのサブ画像番号の画像を表示します	処理ユニット制御	-	-	-	○	P. 8-178
MeasureDispT	処理ユニットの詳細結果を表示します。	処理ユニット制御	-	-	-	○	P. 8-179
MeasureId$	計測 ID を取得します。	処理ユニット制御	○	-	-	○	P. 8-180
MeasureProc	処理ユニットの計測処理を実行します。	処理ユニット制御	○	○	○	○	P. 8-181
MeasureStart	計測トリガの入力を許可します。	計測制御	-	○	○	-	P. 8-182
MeasureStop	計測トリガの入力を禁止します。	計測制御	-	○	○	-	P. 8-183
MemoryInfo	現在のメモリの使用状況（サイズ）を取得します。	その他	○	○	○	○	P. 8-184
MessageBox	メッセージボックスを表示します。	その他	○	○	○	○	P. 8-185
Mid$	文字列の一部を取り出します。	文字列演算	○	○	○	○	P. 8-186
Mkdir	フォルダを作成します。	ファイル制御	○	○	○	○	P. 8-188
MOD	剰余を求めます。	数値演算	○	○	○	○	P. 8-189
MoveUnit	処理ユニットを移動します。	処理フロー制御	-	○	○	-	P. 8-190
NOT	式の否定結果を求めます。	数値演算	○	○	○	○	P. 8-191
Open For Append As#	ファイルを追加書き込みモードで開きます。	ファイル制御	○	○	○	○	P. 8-192
Open For Input As#	ファイルを読み取りモードで開きます。	ファイル制御	○	○	○	○	P. 8-193

関数・コマンド名	機能	種別	ユニット演算マクロ	シーン制御マクロ	通信コマンドマクロ	ユニットマクロ	参照先
Open For Output As#	ファイルを書き込みモードで開きます。	ファイル制御	○	○	○	○	P. 8-194
OpenTextData	メッセージファイルを開きます。	多国語対応メッセージ	○	-	-	○	P. 8-196
Option Explicit	Dim 変数の書式で定義した変数の未定義や重複のチェックを実行します。	その他	-	○	○	○	P. 8-198
OR	2 式の論理和を求めます。	数値演算	○	○	○	○	P. 8-199
Piece$	文字列から区切り文字で区切られた部分文字列を取り出します。	文字列演算	○	○	○	○	P. 8-200
Print	システム状態監視コンソールウィンドウにデータを出力します。	デバッグ	○	○	○	○	P. 8-202
Print#	ファイルにデータを出力します。	ファイル制御	○	○	○	○	P. 8-203
PutAll	すべての出力端子の出力状態を設定します。	入出力制御	-	○	○	○	P. 8-204
PutPort	指定した出力端子の出力状態を設定します。	入出力制御	-	○	○	○	P. 8-205
RaiseErrorProcEvent	UI 画面にエラー処理イベントを通知します。	その他	-	○	○	-	P. 8-207
RaiseOptionEvent	UI 画面にオプションイベントを通知します。	その他	-	○	○	-	P. 8-209
ReadPlcMemory	PLC のメモリ領域から値を読み込みます。	入出力制御	-	○	○	○	P. 8-210
ReceiveData	データを受信します。	入出力制御	-	○	○	○	P. 8-212
ReDim	配列変数を再定義します。	その他	○	○	○	○	P. 8-214
RefreshImageWindow	画像表示ウィンドウの表示を更新します。	表示制御	-	○	○	-	P. 8-215
RefreshJudgeWindow	判定表示ウィンドウの表示を更新します。	表示制御	-	○	○	-	P. 8-216
RefreshTextWindow	テキスト表示ウィンドウの表示を更新します。	表示制御	-	○	○	-	P. 8-217
RefreshTimeWindow	情報表示ウィンドウの表示を更新します。	表示制御	-	○	○	-	P. 8-218
Rem	プログラムに注釈を入れます。	その他	○	○	○	○	P. 8-219
Remeasure	再計測処理を実行します。	計測制御	-	-	○	-	P. 8-220
RenumUnitNo	フロー編集後の処理ユニット番号を取得します。	その他	○	-	-	○	P. 8-221
RGB	カラー値を表す数値を取得します。	数値演算	○	-	-	○	P. 8-222
Right$	文字列の右側から指定した長さの文字列を取り出します。	文字列演算	○	○	○	○	P. 8-223
Rmdir	フォルダを削除します。	ファイル制御	○	○	○	○	P. 8-224
RunOut	RUN 信号の出力状態を設定します。	入出力制御	-	○	○	○	P. 8-225
SaveBackupData	システム+ シーングループ 0 データを保存します。	データセーブ／ロード	-	○	○	-	P. 8-226

関数・コマンド名	機能	種別	ユニット演算マクロ	シーン制御マクロ	通信コマンドマクロ	ユニットマクロ	参照先
SaveData	本体にデータを保存します。	データセーブ／ロード	-	○	○	-	P. 8-227
SaveImage	画像データを保存します。	計測制御	-	○	○	-	P. 8-228
SaveMeasureImage	処理ユニットの計測画像を保存します。	処理ユニット制御	-	-	-	○	P. 8-229
SaveScene	シーンデータを保存します。	データセーブ／ロード	-	○	○	-	P. 8-231
SaveSceneGroup	シーングループデータを保存します。	データセーブ／ロード	-	○	○	-	P. 8-232
SaveSystemData	システムデータを保存します。	データセーブ／ロード	-	○	○	-	P. 8-233
SaveUnitData	処理ユニットデータを保存します。	データセーブ／ロード	-	○	○	-	P. 8-234
SceneCount	使用可能なシーン数を取得します。	シーン制御	-	○	○	-	P. 8-235
SceneDescription$	シーンの説明を取得します。	シーン制御	-	○	○	-	P. 8-236
SceneGroupCount	使用可能なシーングループ数を取得します。	シーングループ制御	-	○	○	-	P. 8-237
SceneGroupNo	現在のシーングループのシーングループ番号を取得します。	シーングループ制御	-	○	○	-	P. 8-238
SceneGroupTitle$	シーングループのタイトル名を取得します。	シーングループ制御	-	○	○	-	P. 8-239
SceneMaker$	シーンの作成者名を取得します。	シーン制御	-	○	○	-	P. 8-240
SceneNo	現在のシーンのシーン番号を取得します。	シーン制御	-	-	○	-	P. 8-241
SceneTitle$	シーンのタイトル名を取得します。	シーン制御	-	○	○	-	P. 8-242
ScreenCapture	画面のキャプチャを保存します。	その他	-	○	○	-	P. 8-243
Select Case Case Else End Select	指定した条件によって処理の流れを制御します。	一般命令	○	○	○	○	P. 8-244
SendData	データを送信します。	入出力制御	-	○	○	○	P. 8-246
SendString	文字列データを送信します。	入出力制御	-	○	○	○	P. 8-248
SetDisplayUnitNo	フロー表示ウィンドウの処理ユニット番号の選択状態を設定します。	表示制御	-	○	○	-	P. 8-249
SetDrawStyle	グラフィック図形の描画属性を設定します。	画像表示ウィンドウ制御	-	-	-	○	P. 8-250
SetForegroundLine	複数ラインランダムトリガモードまたはノンストップ調整モードで、操作対象のラインを指定します。	その他	-	○	○	○	P. 8-252
SetGlobalData	グローバルデータを設定します。	グローバルデータ	○	○	○	○	P. 8-253
SetImageWindow	画像表示ウィンドウの状態を設定します。	表示制御	-	○	○	○	P. 8-254

関数・コマンド名	機能	種別	ユニット演算マクロ	シーン制御マクロ	通信コマンドマクロ	ユニットマクロ	参照先
SetMeasureImage	処理ユニットの計測画像を設定します。	処理ユニット制御	-	-	-	○	P. 8-256
SetMeasureOut	計測結果の外部出力設定を設定します。	計測制御	-	○	○	-	P. 8-257
SetPlcData	WritePlcMemory 関数で書き込むデータを作成します。	入出力制御	-	○	○	○	P. 8-258
SetPollingState	通信モジュールの実行状態を設定します。	入出力制御	-	○	○	○	P. 8-260
SetProfileData	設定ファイル（.ini ファイル）のデータを設定します。	その他	-	○	○	○	P. 8-262
SetSceneData	シーン制御マクロに関するデータやシーン変数の値を設定します。	シーン制御	○	○	○	○	P. 8-263
SetSceneDescription	シーンの説明を設定します。	シーン制御	-	○	○	-	P. 8-265
SetSceneGroupData	指定した識別名のシーングループデータの内容を設定します。	シーングループ制御	-	○	○	-	P. 8-266
SetSceneGroupTitle	シーングループのタイトル名を設定します。	シーングループ制御	-	○	○	-	P. 8-267
SetSceneMaker	シーンの作成者名を設定します。	シーン制御	-	○	○	-	P. 8-268
SetSceneTitle	シーンのタイトル名を設定します。	シーン制御	-	○	○	-	P. 8-269
SetStop	プログラム実行中断の条件を設定します。	デバッグ	○	○	○	○	P. 8-270
SetSystemData	システムデータまたは、システム変数に値を設定します。	システムデータ	○	○	○	○	P. 8-271
SetTextStyle	文字列の描画属性を設定します。	画像表示ウィンドウ制御	-	-	-	○	P. 8-272
SetTextWindow	テキスト表示ウィンドウの状態を設定します。	表示制御	-	○	○	-	P. 8-274
SetUnitData	処理ユニットのデータを設定します。	処理ユニット制御	○	○	○	○	P. 8-275
SetUnitFigure	処理ユニットの図形データを設定します。	処理ユニット制御	-	○	○	○	P. 8-276
SetUnitJudge	処理ユニットの判定結果を設定します。	処理ユニット制御	○	-	-	○	P. 8-277
SetUnitTitle	処理ユニットのタイトル名を設定します。	処理ユニット制御	-	○	○	○	P. 8-278
SetUserSubroutine	外部 DLL に定義されたユーザ定義関数を登録します。	デバッグ	○	○	○	○	P. 8-279
SetVar	指定した変数名の変数を一括で設定します。	その他	○	○	○	○	P. 8-281
Sin	指定した数式の正弦（サイン）を求めます。	数値演算	○	○	○	○	P. 8-283
Sqr	平方根を求めます。	数値演算	○	○	○	○	P. 8-284
StartTimer	経過時間の測定を開始します。	その他	○	○	○	○	P. 8-285
Stop	プログラムの実行を中断します。	デバッグ	○	○	○	○	P. 8-286
Str$	数値を数字文字列に変換します。	文字列演算	○	○	○	○	P. 8-287

関数・コマンド名	機能	種別	ユニット演算マクロ	シーン制御マクロ	通信コマンドマクロ	ユニットマクロ	参照先
Str2$	数値を書式指定して数字文字列に変換します。	文字列演算	○	○	○	○	P. 8-288
Sub ～ End Sub	サブルーチンを定義します。	一般命令	○	○	○	○	P. 8-290
SubList	サブルーチンの一覧をコンソールウィンドウに出力します。	一般命令	○	○	○	○	P. 8-292
SystemInfo	システムに関する情報を取得します。	その他	○	○	○	○	P. 8-293
SystemReset	センサコントローラを再起動します。	その他	-	-	○	-	P. 8-294
Tan	指定した数式の正接（タンジェント）を求めます。	数値演算	○	○	○	○	P. 8-295
TestMeasure	テスト計測を実行します。	計測制御	-	○	○	-	P. 8-296
Time$	内蔵クロックの時刻を読み出します。	数値演算	○	○	○	○	P. 8-297
Timer	経過時間を取得します。	その他	○	○	○	○	P. 8-298
TotalJudge	総合判定結果を取得します。	処理ユニット制御	○	-	-	○	P. 8-300
TransformAngle	角度の値にキャリブレーション結果や位置ずれ修正量を反映します。	処理ユニット制御	○	-	-	○	P. 8-301
TransformArea	面積の値にキャリブレーション結果や位置ずれ修正量を反映します。	処理ユニット制御	○	-	-	○	P. 8-302
TransformDist	距離の値にキャリブレーション結果や位置ずれ修正量を反映します。	処理ユニット制御	○	-	-	○	P. 8-303
TransformLine	直線成分の値にキャリブレーション結果や位置ずれ修正量を反映します。	処理ユニット制御	○	-	-	○	P. 8-304
TransformXY	座標の値にキャリブレーション結果や位置ずれ修正量を反映します。	処理ユニット制御	○	-	-	○	P. 8-305
Try Catch End Try	エラーの発生を検知して例外処理します。	一般命令	○	○	○	○	P. 8-307
UCase$	小文字を大文字に変換します。	文字列演算	○	○	○	○	P. 8-309
UnitCount	処理ユニットの登録数を取得します。	処理フロー制御	-	○	○	-	P. 8-310
UnitData	処理ユニットの数値データを取得します。	処理ユニット制御	○	○	○	○	P. 8-311
UnitData$	処理ユニットの文字列データを取得します。	処理ユニット制御	○	○	○	○	P. 8-312
UnitData2	処理ユニットの描画座標データを取得します。	処理ユニット制御	○	-	-	○	P. 8-313
UnitInfo	処理ユニットの情報を取得します。	処理ユニット制御	○	○	○	○	P. 8-314
UnitItemIdent$	処理ユニットの処理項目識別名を取得します。	処理ユニット制御	○	○	○	○	P. 8-316
UnitJudge	処理ユニットの判定結果を取得します。	処理ユニット制御	○	-	-	○	P. 8-317

関数・コマンド名	機能	種別	ユニット演算マクロ	シーン制御マクロ	通信コマンドマクロ	ユニットマクロ	参照先
UnitNo	処理ユニット番号を取得します。	処理ユニット制御	○	-	-	○	P. 8-318
UnitTitle$	処理ユニットのタイトル名を取得します。	処理ユニット制御	○	○	○	○	P. 8-319
Ut	ユニットラベルを指定して処理ユニット番号を取得します。	シーン制御	○	○	○	○	P. 8-320
Val	数字文字列を数値に変換します。	文字列演算	○	○	○	○	P. 8-321
VarList	システム状態監視コンソールウィンドウに指定した変数の値の一覧を出力します。	デバッグ	○	○	○	○	P. 8-322
Varpop	一時的に退避した変数の値を復元します。	その他	○	○	○	○	P. 8-323
Varpush	変数の値を一時的に退避します。	その他	○	○	○	○	P. 8-325
VarSave	変数の値をシーンデータに保存します。	その他	-	○	-	-	P. 8-327
Wait	指定した時間だけ処理を待機します。	その他	-	○	○	○	P. 8-328
WritePlcMemory	PLC のメモリ領域に値を書き込みます。	入出力制御	-	○	○	○	P. 8-329
XOR	2 式の排他的論理和を求めます。	数値演算	○	○	○	○	P. 8-331

8-1-2　インデックス（機能別）

● 一般命令

関数名	機能	参照先
Call（構造化サブルーチン呼出）	指定したサブルーチンに引数を付加して実行します。	P.8-35
Dim	配列変数を定義します。	P.8-67
Do Loop While	条件を満たしている間、Do から Loop までの間の文を繰り返し実行します。	P.8-71
Erase	配列変数を解放します。	P.8-105
Errcmnd$	エラーが発生したマクロ関数名を取得します。	P.8-106
Errno	エラー番号を取得します。	P.8-107
For To Step Next	For 文と Next 文の間の文を繰り返し実行します。	P.8-118
Gosub	指定したサブルーチンを実行します。	P.8-140
Goto	指定したラベルの行へ処理を移します。	P.8-141
If Then Else	指定した条件によって処理の流れを制御します。	P.8-143
If Then Elseif Else Endlf	指定した条件によって処理の流れを制御します。	P.8-144
Option Explicit	Dim 変数の書式で定義した変数の未定義や重複のチェックを実行します。	P.8-198
Redim	配列変数を再定義します。	P.8-214
Select Case Case Else End Select	指定した条件によって処理の流れを制御します。	P.8-244
Sub ～ End Sub	サブルーチンを定義します。	P.8-290
SubList	サブルーチンの一覧をコンソールウィンドウに出力します。	P.8-292
Try Catch End Try	エラーの発生を検知して例外処理します。	P.8-307

● 数値演算

関数名	機能	参照先
Abs	指定した数式の絶対値を求めます。	P.8-21
AND	指定したふたつの式の論理積を求めます。	P.8-25
ApproximationCircle	近似円を求めます。	P.8-28
Atn	指定した数式の逆正接（アークタンジェント）を求めます。	P.8-33
Cos	指定した数式の余弦（コサイン）を求めます。	P.8-59
Crspoint	2 直線の交点を求めます。	P.8-60
Date$	内蔵クロックの日付を読み出します。	P.8-62
Dposline	指定した直線と点の最短距離を求めます。	P.8-72
Exp	自然対数の底 e を底とする指数関数の値を求めます。	P.8-114
Fix	数値の小数点以下を切り捨てた整数値を求めます。	P.8-117
Int	数値を整数値に変換します。	P.8-153
Log	自然対数値を求めます。	P.8-174
Lsqumeth	複数点の座標から最小二乗法を使って近似直線を求めます。	P.8-175
MOD	剰余を求めます。	P.8-189
NOT	式の否定結果を求めます。	P.8-191
OR	2 式の論理和を求めます。	P.8-199
RGB	カラー値を表す数値を取得します。	P.8-222
Sin	指定した数式の正弦（サイン）を求めます。	P.8-283
Sqr	平方根を求めます。	P.8-284
Tan	指定した数式の正接（タンジェント）を求めます。	P.8-295
Time$	内蔵クロックの時刻を読み出します。	P.8-297
XOR	2 式の排他的論理和を求めます。	P.8-331

● 文字列演算

関数名	機能	参照先
Asc	指定した文字のキャラクタコードの値を求めます。	P.8-31
Chr$	指定したキャラクタコードに対応する文字を求めます。	P.8-41
Hex$	数式の値を 16 進数表現の文字列に変換します。	P.8-142
LCase$	大文字を小文字に変換します。	P.8-162
Left$	文字列の左側から指定した長さの文字列を取り出します。	P.8-163
Len	指定した文字列の長さを求めます。	P.8-164
Mid$	文字列の一部を取り出します。	P.8-186
Piece$	文字列から区切り文字で区切られた部分文字列を取り出します。	P.8-200
Right$	文字列の右側から指定した長さの文字列を取り出します。	P.8-223
Str$	数値を数字文字列に変換します。	P.8-287
Str2$	数値を書式指定して数字文字列に変換します。	P.8-288
UCase$	小文字を大文字に変換します。	P.8-309
Val	数字文字列を数値に変換します。	P.8-321

● シーン制御

関数名	機能	参照先
ChangeScene	シーンを切り替えます。	P.8-38
ClearScene	シーンデータをクリアします。	P.8-43
CopyScene	シーンデータをコピーします。	P.8-52
GetSceneData	シーン制御マクロに関するデータやシーン変数の値を取得します。	P.8-131
SceneCount	使用可能なシーン数を取得します。	P.8-235
SceneDescription$	シーンの説明を取得します。	P.8-236
SceneMaker$	シーンの作成者名を取得します。	P.8-240
SceneNo	現在のシーンのシーン番号を取得します。	P.8-241
SceneTitle$	シーンのタイトル名を取得します。	P.8-242
SetSceneData	シーン制御マクロに関するデータやシーン変数の値を設定します。	P.8-263
SetSceneDescription	シーンの説明を設定します。	P.8-265
SetSceneMaker	シーンの作成者名を設定します。	P.8-268
SetSceneTitle	シーンのタイトル名を設定します。	P.8-269
Ut	ユニットラベルを指定して処理ユニット番号を取得します。	P.8-320

● シーングループ制御

関数名	機能	参照先
ChangeSceneGroup	シーングループを切り替えます。	P.8-39
ClearSceneGroup	シーングループデータをクリアします。	P.8-44
CopySceneGroup	シーングループデータをコピーします。	P.8-53
GetSceneGroupData	指定した識別名のシーングループデータの内容を取得します。	P.8-132
SceneGroupCount	使用可能なシーングループ数を取得します。	P.8-237
SceneGroupNo	現在のシーングループのシーングループ番号を取得します。	P.8-238
SceneGroupTitle$	シーングループのタイトル名を取得します。	P.8-239
SetSceneGroupData	指定した識別名のシーングループデータの内容を設定します。	P.8-266
SetSceneGroupTitle	シーングループのタイトル名を設定します。	P.8-267

● 処理項目制御

関数名	機能	参照先
ItemCount	使用可能な処理項目の種類の数を取得します。	P.8-155
ItemIdent$	処理項目の情報を取得します。	P.8-156
ItemInfo	処理項目の識別名を取得します。	P.8-157
ItemTitle$	処理項目のタイトル名を取得します。	P.8-158

● 処理フロー制御

関数名	機能	参照先
AssignUnit	処理ユニットを登録します。	P.8-32
CheckUnit	処理ユニットの登録状態をチェックします。	P.8-40
CopyUnit	処理ユニットをコピーします。	P.8-54
DeleteUnit	処理ユニットを削除します。	P.8-66
DisableUnit	指定ユニットの計測 ON/OFF を切り替えます。	P.8-68
InsertUnit	処理ユニットを挿入します。	P.8-152
MoveUnit	処理ユニットを移動します。	P.8-190
UnitCount	処理ユニットの登録数を取得します。	P.8-310

● 処理ユニット制御

関数名	機能	参照先
CopyMeasureImage	計測画像をユニットマクロ処理ユニットの画像としてコピーします。	P.8-51
CopyUnitFigure	処理ユニットの図形データをコピーします。	P.8-55
CopyUnitImage	処理ユニットの画像をユニットマクロ処理ユニットの画像としてコピーします。	P.8-56
CopyUnitModel	処理ユニットのモデルデータをコピーします。	P.8-57
GetImageSize	処理ユニットの画像の画像サイズを取得します。	P.8-122
GetUnitData	処理ユニットのデータを取得します。	P.8-138
GetUnitFigure	処理ユニットの図形データを取得します。	P.8-139
ImageFormat	処理ユニットの画像の画像フォーマットを取得します。	P.8-146
MeasureDispG	処理ユニットの計測結果の表示処理を実行します。	P.8-177
MeasureDispI	処理ユニットのサブ画像番号の画像を表示します	P.8-178
MeasureDispT	処理ユニットの詳細結果を表示します。	P.8-179
MeasureId$	計測 ID を取得します。	P.8-180
MeasureProc	処理ユニットの計測処理を実行します。	P.8-181
SaveMeasureImage	処理ユニットの計測画像を保存します。	P.8-229
SetMeasureImage	処理ユニットの計測画像を設定します。	P.8-256
SetUnitData	処理ユニットのデータを設定します。	P.8-275
SetUnitFigure	処理ユニットの図形データを設定します。	P.8-276
SetUnitJudge	処理ユニットの判定結果を設定します。	P.8-277
SetUnitTitle	処理ユニットのタイトル名を設定します。	P.8-278
TotalJudge	総合判定結果を取得します。	P.8-300
TransformAngle	角度の値にキャリブレーション結果や位置ずれ修正量を反映します。	P.8-301
TransformArea	面積の値にキャリブレーション結果や位置ずれ修正量を反映します。	P.8-302
TransformDist	距離の値にキャリブレーション結果や位置ずれ修正量を反映します。	P.8-303
TransformLine	直線成分の値にキャリブレーション結果や位置ずれ修正量を反映します。	P.8-304

関数名	機能	参照先
TransformXY	座標の値にキャリブレーション結果や位置ずれ修正量を反映します。	P.8-305
UnitData	処理ユニットの数値データを取得します。	P.8-311
UnitData$	処理ユニットの文字列データを取得します。	P.8-312
UnitData2	処理ユニットの描画座標データを取得します。	P.8-313
UnitInfo	処理ユニットの情報を取得します。	P.8-314
UnitItemIdent$	処理ユニットの処理項目識別名を取得します。	P.8-316
UnitJudge	処理ユニットの判定結果を取得します。	P.8-317
UnitNo	処理ユニット番号を取得します。	P.8-318
UnitTitle$	処理ユニットのタイトル名を取得します。	P.8-319

● 計測制御

関数名	機能	参照先
ClearMeasureData	処理ユニットの計測結果をクリアします。	P.8-42
ContinuousMeasure	連続計測の開始／停止を行います。	P.8-50
GetMeasureOut	計測結果の外部出力設定を取得します。	P.8-125
ImageUpdate	カメラからの入力画像を更新します。	P.8-147
Measure	計測処理を実行します。	P.8-176
MeasureStart	計測トリガの入力を許可します。	P.8-182
MeasureStop	計測トリガの入力を禁止します。	P.8-183
Remeasure	再計測処理を実行します。	P.8-220
SaveImage	画像データを保存します。	P.8-228
SetMeasureOut	計測結果の外部出力設定を設定します。	P.8-257
TestMeasure	テスト計測を実行します。	P.8-296

● 入出力制御

関数名	機能	参照先
BusyOut	処理実行中信号の出力状態を設定します。	P.8-34
ErrorOut	ERROR 信号の出力状態を設定します。	P.8-108
GetAll	すべての入力端子の入力状態を取得します。	P.8-120
GetPlcData	ReadPlcMemory 関数で読み込んだデータを取り出します。	P.8-126
GetPollingState	通信モジュールの実行状態を取得します。	P.8-127
GetPort	指定した入力端子の入力状態を取得します。	P.8-129
JudgeOut	総合判定信号の出力状態を設定します。	P.8-159
PutAll	すべての出力端子の出力状態を設定します。	P.8-204
PutPort	指定した出力端子の出力状態を設定します。	P.8-205
ReadPlcMemory	PLC のメモリ領域から値を読み込みます。	P.8-210
ReceiveData	データを受信します。	P.8-212
RunOut	RUN 信号の出力状態を設定します。	P.8-225
SendData	データを送信します。	P.8-246
SendString	文字列データを送信します。	P.8-248
SetPlcData	WritePlcMemory 関数で書き込むデータを作成します。	P.8-258
SetPollingState	通信モジュールの実行状態を設定します。	P.8-260
WritePlcMemory	PLC のメモリ領域に値を書き込みます。	P.8-329

● 表示制御

関数名	機能	参照先
DisplayUnitNo	フロー表示ウィンドウの処理ユニット番号の選択状態を取得します。	P.8-70
GetImageWindow	画像表示ウィンドウの状態を取得します。	P.8-123
GetTextWindow	テキスト表示ウィンドウの状態を取得します。	P.8-137
RefreshImageWindow	画像表示ウィンドウの表示を更新します。	P.8-215
RefreshJudgeWindow	判定表示ウィンドウの表示を更新します。	P.8-216
RefreshTextWindow	テキスト表示ウィンドウの表示を更新します。	P.8-217
RefreshTimeWindow	情報表示ウィンドウの表示を更新します。	P.8-218
SetDisplayUnitNo	フロー表示ウィンドウの処理ユニット番号の選択状態を設定します。	P.8-249
SetImageWindow	画像表示ウィンドウの状態を設定します。	P.8-254
SetTextWindow	テキスト表示ウィンドウの状態を設定します。	P.8-274

● 画像表示ウィンドウ制御

関数名	機能	参照先
DisplaySubNo	画像表示サブ番号を取得します。	P.8-69
DrawArc	画像表示ウィンドウに円弧を描画します。	P.8-74
DrawArcW	画像表示ウィンドウに幅付き円弧を描画します。	P.8-75
DrawBox	画像表示ウィンドウに長方形を描画します。	P.8-77
DrawCircle	画像表示ウィンドウに円形を描画します。	P.8-79
DrawCircleW	画像表示ウィンドウに幅付き円形を描画します。	P.8-80
DrawCursor	画像表示ウィンドウに十字カーソルを描画します。	P.8-82
DrawEllipse	画像表示ウィンドウに楕円形を描画します。	P.8-83
DrawFigure	画像表示ウィンドウに図形を描画します。	P.8-85
DrawFillImage	画像表示ウィンドウに塗り潰し画像を描画します。	P.8-86
DrawLine	画像表示ウィンドウに直線を描画します。	P.8-88
DrawLineW	画像表示ウィンドウに幅付き直線を描画します。	P.8-90
DrawMeasureImage	画像表示ウィンドウに計測画像を描画します。	P.8-91
DrawPoint	画像表示ウィンドウに点を描画します。	P.8-92
DrawPolygon	画像表示ウィンドウに多角形を描画します。	P.8-93
DrawSearchFigure	画像表示ウィンドウにサーチ図形を描画します。	P.8-95
DrawTextG	画像表示ウィンドウに文字列を描画します。	P.8-99
DrawUnitImage	画像表示ウィンドウに処理ユニット画像を描画します。	P.8-100
SetDrawStyle	グラフィック図形の描画属性を設定します。	P.8-250
SetTextStyle	文字列の描画属性を設定します。	P.8-272

● テキスト表示ウィンドウ制御

関数名	機能	参照先
DrawJudgeText	テキスト表示ウィンドウに判定結果の文字列を描画します。	P.8-87
DrawText	テキスト表示ウィンドウに文字列を描画します。	P.8-97

● システムデータ

関数名	機能	参照先
AddSystemData	システムデータを追加します。	P.8-24
GetSystemData	システムデータまたは、システム変数の値を取得します。	P.8-133
SetSystemData	システムデータまたは、システム変数に値を設定します。	P.8-271

● グローバルデータ

関数名	機能	参照先
AddGlobalData	グローバルデータを追加します。	P.8-23
GetGlobalData	グローバルデータを取得します。	P.8-121
SetGlobalData	グローバルデータを設定します。	P.8-253

● データセーブ／ロード

関数名	機能	参照先
LoadBackupData	システム+シーングループ 0 データを読み込みます。	P.8-168
LoadScene	シーンデータを読み込みます。	P.8-169
LoadSceneGroup	シーングループデータを読み込みます。	P.8-170
LoadSystemData	システムデータを読み込みます。	P.8-171
LoadUnitData	処理ユニットデータを読み込みます。	P.8-172
SaveBackupData	システム+ シーングループ 0 データを保存します。	P.8-226
SaveData	本体にデータを保存します。	P.8-227
SaveScene	シーンデータを保存します。	P.8-231
SaveSceneGroup	シーングループデータを保存します。	P.8-232
SaveSystemData	システムデータを保存します。	P.8-233
SaveUnitData	処理ユニットデータを保存します。	P.8-234

● ファイル制御

関数名	機能	参照先
Close	ファイルを閉じます。	P.8-45
Dskf	ドライブの空き容量を求めます。	P.8-102
Eof	ファイルの終わりを調べます。	P.8-104
Fcopy	ファイルをコピーします。	P.8-115
Input#	ファイルからデータを読み込みます。	P.8-149
Input$	ファイルからバイナリデータを読み込みます。	P.8-150
Isfile	ファイルの属性と有無を調べます。	P.8-154
Kill	ファイルまたはフォルダを削除します。	P.8-160
Line Input#	ファイルから 1 行分のデータを読み込みます。	P.8-165
Mkdir	フォルダを作成します。	P.8-188
Open For Append As#	ファイルを追加書き込みモードで開きます。	P.8-192
Open For Input As#	ファイルを読み取りモードで開きます。	P.8-193
Open For Output As#	ファイルを書き込みモードで開きます。	P.8-194
Print#	ファイルにデータを出力します。	P.8-203
Rmdir	フォルダを削除します。	P.8-224

● 多言語対応メッセージ

関数名	機能	参照先
CloseTextData	メッセージファイルを閉じます。	P.8-46
GetText	メッセージファイルからテキストデータを取得します。	P.8-134
OpenTextData	メッセージファイルを開きます。	P.8-196

● デバッグ

関数名	機能	参照先
Call（ユーザ定義関数実行）	登録されているユーザ定義関数を実行します。	P.8-36

関数名	機能	参照先
Cont	プログラムの実行を中断した状態からプログラムの実行を再開します。	P.8-49
Debug	プログラムの実行形態と情報出力方法を設定します。	P.8-63
DebugPrint	システム状態監視コンソールウィンドウにデバッグ情報を出力します。	P.8-64
List	システム状態監視コンソールウィンドウにプログラム内容の全部または一部を出力します。	P.8-167
Print	システム状態監視コンソールウィンドウにデータを出力します。	P.8-202
SetStop	プログラム実行中断の条件を設定します。	P.8-270
SetUserSubroutine	外部 DLL に定義されたユーザ定義関数を登録します。	P.8-279
Stop	プログラムの実行を中断します。	P.8-286
VarList	システム状態監視コンソールウィンドウに指定した変数の値の一覧を出力します。	P.8-322

● その他

関数名	機能	参照先
AbsolutePath$	クイックアクセスの識別名に割り当てられた絶対パスを取得します。	P.8-22
ApplicationPath$	アプリケーションソフトに関連したパス名を取得します。	P.8-26
ApplicationVersion$	アプリケーションソフトのバージョン情報を取得します。	P.8-27
ArrayDims	配列の次元数を取得します。	P.8-29
ArrayLen	配列の要素数を取得します。	P.8-30
Cls	コンソールウィンドウの表示をクリアします。	P.8-48
DeleteProfileData	設定ファイル（.ini ファイル）のデータを削除します。	P.8-65
ElapsedTime	計測開始からの経過時間を取得します。	P.8-103
ExecuteErrorProc	指定した種別のエラーが発生したことをシステムに通知します。	P.8-110
ExecuteImageLogging	画像ロギングを実行します。	P.8-112
ExitFzProcess	センサコントローラを終了させます。	P.8-113
FigureType	処理ユニットに設定可能な図形種類を取得します。	P.8-116
GetProfileData	設定ファイル（.ini ファイル）のデータを取得します。	P.8-130
LanguageString$	現在の言語設定にしたがった言語で文字列を取得します。	P.8-161
MemoryInfo	現在のメモリの使用状況（サイズ）を取得します。	P.8-184
MessageBox	メッセージボックスを表示します。	P.8-185
RaiseErrorProcEvent	UI 画面にエラー処理イベントを通知します。	P.8-207
RaiseOptionEvent	UI 画面にオプションイベントを通知します。	P.8-209
Rem	プログラムに注釈を入れます。	P.8-219
RenumUnitNo	フロー編集後の処理ユニット番号を取得します。	P.8-221
ScreenCapture	画面のキャプチャを保存します。	P.8-243
SetForegroundLine	複数ラインランダムトリガモードまたはノンストップ調整モードで、操作対象のラインを指定します。	P.8-252
SetProfileData	設定ファイル（.ini ファイル）のデータを設定します。	P.8-262
SetVar	指定した変数名の変数を一括で設定します。	P.8-281
StartTimer	経過時間の測定を開始します。	P.8-285
SystemInfo	システムに関する情報を取得します。	P.8-293
SystemReset	センサコントローラを再起動します。	P.8-294
Timer	経過時間を取得します。	P.8-298
VarPop	一時的に退避した変数の値を復元します。	P.8-323
VarPush	変数の値を一時的に退避します。	P.8-325
VarSave	変数の値をシーンデータに保存します。	P.8-327

関数名	機能	参照先
Wait	指定した時間だけ処理を待機します。	P.8-328

8-2 コマンド詳細

Abs

指定した数式の絶対値を求めます。

・書式

Abs(<expression>)

・パラメータ

パラメータ名	データ型	説明
<expression>	整数型 実数型	絶対値を求める数式

・戻り値

実数型の絶対値の値を返します。

・解説

<expression> パラメータで指定した数式の絶対値を求めます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

2 点 (X1,Y1), (X2,Y2) の各座標の差を求めます。

```
X1# = 100
Y1# = 200
X2# = 200
Y2# = 100

DX# = Abs(X1# - X2#)
DY# = Abs(Y1# - Y2#)
```

結果は以下のとおりになります。

```
DX# = 100
DY# = 100
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「GetUnitData (P.8-138)」
「UnitData (P.8-311)」

AbsolutePath$

クイックアクセスの識別名に割り当てられた絶対パスを取得します。

・書式

AbsolutePath$(<pathName>)

・パラメータ

パラメータ名	データ型	説明
<pathName>	文字列型	クイックアクセスの識別名 クイックアクセスについては、『画像処理システム FH/FHV シリーズ ユーザーズマニュアル (SDNB-719)』の「よく使用するフォルダを登録して簡単にアクセスできるようにする［クイックアクセス設定ツール］」を参照してください。

・戻り値

絶対パス名を返します。

・解説

<pathName>パラメータで指定したクイックアクセスの識別名に割り当てられた絶対パスを取得します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

［クイックアクセス設定ツール］でフォルダに設定した［識別名］Logging の［絶対パス］を取得して、フォルダにアクセスします。

```
LoggingPATH$ = AbsolutePath$("<Logging>")
testmeasure LoggingPATH$ + "test.ifz"
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 6.40 以降

・関連項目

なし

AddGlobalData

グローバルデータを追加します。

・書式

AddGlobalData <dataIdent>, <data>

・パラメータ

パラメータ名	データ型	説明
<dataIdent>	文字列型	追加するグローバルデータの識別名
<data>	整数型 実数型 文字列型	グローバルデータを追加した際に設定する値

・戻り値

なし

・解説

<dataIdent> パラメータで指定した識別名のデータをグローバルデータに追加して、追加したデータに<data> パラメータで指定した値を設定します。
指定した識別名のグローバルデータがすでに存在する場合は、何もせず終了します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
<dataIdent> パラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

識別名が"ABC" のグローバルデータを追加して、値として 1 を設定します。

```
Rem グローバルデータ"ABC" を追加し、値として 1 を設定します
AddGlobalData "ABC", 1

Rem グローバルデータ"ABC" に設定された値（整数値）を取得し、変数 DATA& に格納します
GetGlobalData "ABC", DATA&
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「GetGlobalData (P.8-121)」
「SetGlobalData (P.8-253)」

AddSystemData

システムデータを追加します。

・書式

AddSystemData <dataIdent0>, <dataIdent1>, <data>

・パラメータ

パラメータ名	データ型	説明
<dataIdent0>	文字列型	追加するシステムデータの識別情報 0 のデータ識別名（"PanDA" を指定）
<dataIdent1>	文字列型	追加するシステムデータの識別情報 1 のデータ識別名
<data>	整数型 実数型 文字列型	追加するシステムデータの値

・戻り値

なし

・解説

<dataIdent0> パラメータで指定した識別情報 0 に属する、<dataIdent1> パラメータで指定した識別情報 1 のデータをシステムデータに追加して、追加したデータに<data> パラメータで指定した値を設定します。
<dataIdent0> パラメータに"PanDA" 以外を指定した場合は、Illegal function call のエラーが発生します。
指定した識別情報 0 に属する識別情報 1 のシステムデータがすでに登録されている場合は、何もせず終了します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
<dataIdent0> パラメータとして、存在しない識別名を指定した場合は、Illegal function call のエラーが発生します。
<dataIdent1> パラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

識別情報 0 が"PanDA" のシステムデータに、識別情報 1 が"LoggingCount" のデータを追加します。設定データは 20 を設定します。

```
AddSystemData "PanDA", "LoggingCount", 20
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「AddGlobalData (P.8-23)」
「GetSystemData (P.8-133)」
「SetSystemData (P.8-271)」

AND

指定したふたつの式の論理積を求めます。

・書式

<expression1> AND <expression2>

・パラメータ

パラメータ名	データ型	説明
<expression1>	整数型	論理積を求める数式
<expression2>	整数型	

・戻り値

整数型の論理積の値を返します。

・解説

<expression1> パラメータで指定した数式と、<expression2> パラメータで指定した数式のビットごとの論理積を求めます。
<expression1> パラメータと<expression2> パラメータの値が実数型の場合、小数部を四捨五入した値に対する論理積を求めます。
If 文の中で And 条件としても使用できます。論理式の詳細については、「4-1-5 演算子 (P.4-12)」を参照してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

変数 X と変数 Y の値の論理積を求めます。

```
X& = 15
Y& = 8

DATA& = X& AND Y&
```

結果は以下のとおりになります。

```
DATA& = 8
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「GetUnitData (P.8-138)」
「NOT (P.8-191)」
「OR (P.8-199)」
「UnitData (P.8-311)」
「XOR (P.8-331)」

ApplicationPath$

アプリケーションソフトに関連したパス名を取得します。

・書式

ApplicationPath$(<kind>)

・パラメータ

パラメータ名	データ型	説明
<kind>	文字列型	取得したいアプリケーションソフトに関連するパス名の種別です。 0：アプリケーションソフトがインストールされたパス名 1：ファイル操作の基準パス名 2：データ保存先のデフォルトパス名 3：システムデータ保存先パス名 10000 + シーングループ番号：該当する番号のシーングループデータの保存先パス名

・戻り値

<kind> で指定したパス名情報（文字列）を返します。

・解説

パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定しても、Illegal function call のエラーは発生しません。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

無手順通信で画像データを登録画像として登録します。

```
Rem 登録画像の保存先パス取得
USBPATH$ = ApplicationPath$(2)
FILENAME$ = ARGUMENTSTRING$(0)
DATAPATH$ = USBPATH$ + "RegisteredImage¥" + FILENAME$ + ".ifz"

Rem 最新入力画像を登録画像に登録
SaveImage -1, DATAPATH$
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 5.40 以降

・関連項目

「SaveImage (P.8-228)」

ApplicationVersion$

アプリケーションソフトのバージョン情報を取得します。

・書式

ApplicationVersion$(<kind>)

・パラメータ

パラメータ名	データ型	説明
<kind>	整数型	取得したいアプリケーションソフトのバージョン情報の種別です。 0：機種情報 1：バージョン情報 2：作成日付

・戻り値

<kind> で指定したアプリケーションソフトバージョン情報（文字列）を返します。

・解説

アプリケーションソフトのバージョン情報を取得します。
アプリケーションソフト内の Version.ini ファイルの記述内容にしたがった情報が返されます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定しても、Illegal function call のエラーは発生しません。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

アプリケーションソフトのバージョン情報を UDP 無手順で送信します。

```
Rem システムのバージョン情報を取得する
APPVER$ = ApplicationVersion$(1)
SendString "UdpNormal", APPVER$
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 5.40 以降

・関連項目

「SendString (P.8-248)」

ApproximationCircle

近似円を求めます。

・書式

ApproximationCircle <count>, <x()>, <y()>, <centerX>, <centerY>, <radius>

・パラメータ

パラメータ名	データ型	説明
<count>	整数型	指定する座標の個数
<x()>	整数型配列 実数型配列	各点の X 座標を格納した 1 次元配列
<y()>	整数型配列 実数型配列	各点の Y 座標を格納した 1 次元配列
<centerX>	実数型	近似円の中心 X 座標
<centerY>	実数型	近似円の中心 Y 座標
<radius>	実数型	近似円の半径

・戻り値

なし

・解説

<count> パラメータで指定した個数の、<x()> パラメータと<y()> パラメータで指定した座標をもとに近似円を求めます。<centerX> パラメータと<centerY> パラメータには、求めた近似円の中心 X 座標と中心 Y 座標をそれぞれ格納する変数を指定してください。<radius> パラメータには、求めた近似円の半径を格納する変数を指定してください。

<count> パラメータには、3 以上の整数の値を指定してください。

<x()> パラメータと<y()> パラメータには、<count> パラメータで指定した以上の個数の座標値を格納した 1 次元の整数型配列変数または実数型配列変数を、要素番号を指定せず() だけ付加して指定してください。

パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。

パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。

マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

3 点(50, 50)、(100, 100)、(150, 50) より近似円を算出して中心座標と半径を求めます。

```
Dim X&(3), Y&(3)

Rem 3 点の座標値を配列に代入します
X&(0) = 50
Y&(0) = 50
X&(1) = 100
Y&(1) = 100
X&(2) = 150
Y&(2) = 50

Rem 近似円を算出して中心座標と半径を求めます
ApproximationCircle 3, X&(), Y&(), CENTERX#, CENTERY#, RADIUS#
```

・使用可能なモジュール
ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「GetUnitData (P.8-138)」
「Lsqumeth (P.8-175)」
「UnitData (P.8-311)」

ArrayDims

配列の次元数を取得します。

・書式
ArrayDims(<arrayName()>)

・パラメータ

パラメータ名	データ型	説明
<arrayName()>	-	対象の配列の変数名

・戻り値
配列の次元数を返します。

・解説
<arrayName()>パラメータで指定した配列の次元数を取得します。
<arrayName()>パラメータとして、存在しない配列名を指定した場合は、Undefined array のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
このマクロ関数は、シーン変数およびシステム変数には使用できません。

・使用例
配列の次元数を取得します。

```
Dim AA&(1,1,1)

DIMS& = ArrayDims(AA&())
```

結果は以下のとおりになります。

```
DIMS& = 3
```

・使用可能なモジュール
シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 6.40 以降

・関連項目

「ArrayLen (P.8-30)」

ArrayLen

配列の要素数を取得します。

・書式

ArrayLen(<arrayName()>[, <dimension>])

・パラメータ

パラメータ名	データ型	説明
<arrayName()>	-	対象の配列の変数名
<dimension>	整数型	要素数を取得する配列の次元番号

・戻り値

配列の要素数を返します。

・解説

<arrayName()>パラメータで指定した配列の、<dimension>パラメータで指定した次元番号の配列要素数を取得します。
<dimension>パラメータを省略した場合、すべての次元の要素数が取得されます。
<arrayName()>パラメータとして、存在しない配列名を指定した場合は、Undefined array のエラーが発生します。
<arrayName()>パラメータとして、存在しない配列名を指定した場合は、Undefined array のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

このマクロ関数は、シーン変数およびシステム変数には使用できません。

・使用例

配列の要素数を取得します。

```
Dim AA&(1,2)

LEN0& = ArrayLen(AA&(),0)
LEN1& = ArrayLen(AA&(),1)
```

結果は以下のとおりになります。

```
LEN0& = 2
LEN1& = 3
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 6.40 以降

・関連項目

「ArrayDims (P.8-29)」

Asc

指定した文字のキャラクタコードの値を求めます。

・書式

Asc <string>

・パラメータ

パラメータ名	データ型	説明
<string>	文字列型	キャラクタコードを求める文字列

・戻り値

整数型のキャラクタコードの値を 10 進数で返します。

・解説

<string> パラメータで指定した文字列の先頭 1 文字のキャラクタコードを ASCII コードで求めます。
Asc 関数の逆関数には Chr$ 関数があります。Chr$ 関数は、指定したキャラクタコードに対応するキャラクタ文字を返します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定しても、エラーは発生しません。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

文字「A」のキャラクタコードを求めます。

```
CHARA$ = "A"

CODE& = Asc(CHARA$)
```

結果は以下のとおりになります。

```
CODE& = 65
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「Chr$ (P.8-41)」
「LCase$ (P.8-162)」
「Len (P.8-164)」
「Piece$ (P.8-200)」
「Str$ (P.8-287)」
「UCase$ (P.8-309)」
「Hex$ (P.8-142)」
「Left$ (P.8-163)」
「Mid$ (P.8-186)」
「Right$ (P.8-223)」
「Str2$ (P.8-288)」
「Val (P.8-321)」

AssignUnit

処理ユニットを登録します。

・書式

AssignUnit <unitNo>, <itemIdent>

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	処理ユニットを登録する計測フロー上の処理ユニット番号（0～（現在シーンの処理ユニット数））
<itemIdent>	文字列型	登録する処理ユニットの処理項目識別名「9-5 処理項目識別名一覧 (P.9-64)」

・戻り値

なし

・解説

<unitNo> パラメータで指定した計測フロー上の位置に、<itemIdent> パラメータで指定した識別名の処理項目を処理ユニットとして登録します。
<unitNo> パラメータで指定した位置に、すでに処理ユニットが登録されている場合は、上書き登録されます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
<itemIdent> パラメータとして、存在しない識別名を指定した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）

・使用例

計測フローの末尾にサーチの処理ユニットを登録します。

```
Rem 現在の計測フローに登録されている処理ユニット数を取得します
UNUM& = UnitCount

Rem サーチの処理ユニットを追加します
AssignUnit UNUM&, "Search"
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「CheckUnit (P.8-40)」
「CopyUnit (P.8-54)」
「DeleteUnit (P.8-66)」
「InsertUnit (P.8-152)」
「MeasureStart (P.8-182)」
「MeasureStop (P.8-183)」
「MoveUnit (P.8-190)」
「UnitCount (P.8-310)」

Atn

指定した数式の逆正接（アークタンジェント）を求めます。

・書式

Atn(<expression>)

・パラメータ

パラメータ名	データ型	説明
<expression>	整数型 実数型	逆正接を求める数式

・戻り値

実数型の逆正接の値を-π/2 から π/2 までの範囲のラジアンで返します。

・解説

<expression> パラメータで指定した数式の逆正接を求めます。
<expression> パラメータには値をラジアン値で指定します。角度値をラジアン値に変換するには、角度値に π/180 をかけます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

変数 X# の逆正接を求めます。

```
X# = 1

XX# = Atn(X#)*180/3.141592
```

結果は以下のとおりになります。
（小数点第 4 位で四捨五入）

```
XX# = 45.000
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「Cos (P.8-59)」
「Sin (P.8-283)」
「UnitData (P.8-311)」
「GetUnitData (P.8-138)」
「Tan (P.8-295)」

BusyOut

処理実行中信号の出力状態を設定します。

・書式

BusyOut <ioIdent>, <state>

・パラメータ

パラメータ名	データ型	説明
<ioIdent>	文字列型	使用する通信モジュールの識別名 （"Parallello" または "EtherCAT"） （参照：「9-4 IO モジュール一覧 (P.9-25)」）
<state>	整数型	端子の出力状態 0：出力 OFF 状態 1：出力 ON 状態

・戻り値

なし

・解説

<ioIdent> パラメータで指定した通信モジュールの BUSY 信号をはじめとした処理実行中信号に、<state> パラメータで指定した出力状態を設定します。
<ioIdent> パラメータには、通常は"Parallello" または"EtherCAT" を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
<ioIdent> パラメータとして、存在しない識別名を指定した場合は、Illegal function call のエラーが発生します。
<state> パラメータとして、存在しない出力状態の値を指定しても、エラーは発生しません。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

通信コマンドマクロで、パラレル I/O の BUSY 信号を ON に設定します。

```
IOMODULE$ = "Parallello"

Rem 出力状態を設定します
BusyOut IOMODULE$, 1
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「GetAll (P.8-120)」
「GetPort (P.8-129)」
「JudgeOut (P.8-159)」
「PutAll (P.8-204)」
「PutPort (P.8-205)」
「RunOut (P.8-225)」

Call（構造化サブルーチン呼出）

指定したサブルーチンに引数を付加して実行します。

・書式

Call <label>([, <argument0>...<argumentN>])

・パラメータ

パラメータ名	データ型	説明
<label>	文字列型	実行するサブルーチンのラベル名
<argument0>	整数型 実数型 文字列型 配列型 バリアント型	サブルーチンに渡すデータ
:	:	:
<argumentN>	整数型 実数型 文字列型 配列型 バリアント型	サブルーチンに渡すデータ

・戻り値

なし

・解説

<label> パラメータで指定したラベル名の引数付きサブルーチンを <argument0> ～ <argumentN> パラメータを引数として実行します。
サブルーチンの処理の最後の End Sub 文で呼び出し元のサブルーチンに復帰して、このマクロ関数を実行した次の行から処理を再開します。
パラメータとして存在しないラベルを指定した場合は Undefined label エラーが発生します。
引数付きサブルーチンとして定義されている引数の数と、<argument0> ～ <argumentN> パラメータで指定した引数の数に違いがあってもエラーは発生しません。
Call 関数で指定した引数が、引数付サブルーチンで定義されている引数の数よりも多い場合は、余分な引数は無視されます。本関数で指定した引数の方が引数付き差ブルーチンの定義されている引数の数よりも少ない場合は指定されなかった引数にはその変数の型毎の初期値が渡されます。
引数の数が 0 の場合でも()を省略する事はできません。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

ユニットマクロの*MeasureProc サブルーチンで別に定義した*ValueInit サブルーチンの処理を実行します。

```
*MCRINIT
  Dim X#(0)
  Dim Y#(0)
  Dim ANG#(0)
  Dim JG&
Return
*MEASUREPROC
  Rem 他のサブルーチンを実行します
  Call *ValueInit(X#(), Y#(), ANG#(), JG&)
Return
Rem 各種計測値を初期化する
Sub *ValueInit(%POSX#(), %POSY#(), %ANGLE#(), %JUDGE&)
  ReDim %POSX#(10)
  ReDim %POSY#(10)
  ReDim %ANGLE#(10)
  %JUDGE& = JUDGE_NC
End Sub
```

・使用可能なモジュール
ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 6.50 以降

・関連項目

「Sub ～ End Sub (P.8-290)」

Call（ユーザ定義関数実行）

登録されているユーザ定義関数を実行します。

・書式
Call <subroutineIdent>[;¦, <argument>...]

・パラメータ

パラメータ名	データ型	説明
<subroutineIdent>	文字列型	登録されているユーザ定義関数の識別名
<argument>	整数型 実数型 文字列型	登録されているユーザ定義関数の引数

・戻り値
なし

・解説
<subroutineIdent> パラメータで指定した識別名を持つユーザ定義関数を呼び出して実行します。
ユーザ定義関数が引数を持つ場合は、<argument> パラメータで指定したデータが引数としてユーザ定義関数に渡されます。
<argument> パラメータには、ユーザ定義関数の定義内容にしたがって引数を指定してください。
この関数を実行する前に、SetUserSubroutine 関数でユーザ定義関数を登録してください。
ユーザ定義関数の実行内容は、外部 DLL で定義された処理内容にしたがいます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして登録されていないユーザ定義関数の識別名を指定した場合は、Illegal function call のエラーが発生します。

文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
ユーザ定義関数の処理の結果としてエラーが返された場合は、Call 関数の実行結果としてそのエラー種別が表示されます。表示される可能性があるものは以下のとおりです。
- Syntax error
- Illegal function call
- Out of memory
- Type missmatch

この関数で呼び出したユーザ定義関数が、マクロ関数として正しいインタフェースで実装されていなかった場合は、ユーザ定義関数の処理の中でエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
- このマクロ関数では、正しいインタフェースでプログラミングされた DLL のユーザ定義関数しか使用できません。それ以外の DLL やユーザ定義関数を使用した場合は、計測異常やセンサコントローラの異常終了など予期せぬ動作が発生する可能性があります。ユーザ定義関数の作成には FH-AP1 を参照してください。
- このマクロ関数で呼び出すユーザ定義関数の処理の内容によっては、計測異常やセンサコントローラの異常終了など予期せぬ動作が発生する可能性があります。あらかじめ外部装置との接続をはずした状態でセンサコントローラ単体での十分な動作確認とデバッグを実施してください。
- このマクロ関数を、外部装置を接続した実環境で動作させる場合は、必ず外部でフェイルセーフ対策を施してください。

・使用例
MacroUserProc.dll に定義されているユーザ定義関数"UserProc0" を、識別名"USR" として登録します。その後登録したユーザ定義関数を、識別名を指定して呼び出して実行します。

```
Rem ユーザ定義関数をプログラムの中で使えるように登録します
SetUserSubroutine "USR", "MacroUserProc", "UserProc0"

Rem 登録したユーザ定義関数を呼び出して実行します
Call "USR", 0
```

・使用可能なモジュール
ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 5.20 以降

・関連項目

「SetUserSubroutine (P.8-279)」

ChangeScene

シーンを切り替えます。

・書式

ChangeScene <sceneNo>

・パラメータ

パラメータ名	データ型	説明
<sceneNo>	整数型	切替先のシーン番号（0～127）

・戻り値

なし

・解説

現在のシーンを、<sceneNo> パラメータで指定したシーン番号のシーンに切り替えます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）

・使用例

現在のシーンをシーン番号 2 のシーンに切り替えます。

```
ChangeScene 2
```

・使用可能なモジュール

通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「ChangeSceneGroup (P.8-39)」
「MeasureStart (P.8-182)」
「MeasureStop (P.8-183)」
「SceneCount (P.8-235)」
「SceneGroupNo (P.8-238)」
「SceneNo (P.8-241)」

ChangeSceneGroup

シーングループを切り替えます。

・書式

ChangeSceneGroup <sceneGroupNo>, <sceneNo>

・パラメータ

パラメータ名	データ型	説明
<sceneGroupNo>	整数型	切替先のシーングループ番号（0～31）
<sceneNo>	整数型	切替先のシーン番号（0～127）

・戻り値

なし

・解説

現在のシーングループを、<sceneGroupNo> パラメータで指定したシーングループ番号のシーングループの、<sceneNo> パラメータで指定したシーン番号のシーンに切り替えます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）

・使用例

現在のシーングループをシーングループ番号 10 のシーングループのシーン番号 2 のシーンに切り替えます。

```
ChangeSceneGroup 10, 2
```

・使用可能なモジュール

通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「ChangeScene (P.8-38)」
「MeasureStop (P.8-183)」
「SceneGroupNo (P.8-238)」
「MeasureStart (P.8-182)」
「SceneGroupCount (P.8-237)」

CheckUnit

処理ユニットの登録状態をチェックします。

・書式

CheckUnit(<unitNo>)

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	登録状態をチェックする処理ユニットの処理ユニット番号（0～（現在シーンの処理ユニット数 -1））

・戻り値

整数型の登録状態の値を返します。

- 0：処理ユニット未登録
- 1：処理ユニット登録済

・解説

<unitNo> パラメータで指定した計測フロー上の位置に、処理ユニットが登録されているかどうかをチェックします。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定しても、Illegal function call のエラーは発生しません。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

計測フローの処理ユニット番号 3 に処理ユニットが登録されている場合、処理ユニットのタイトル名を取得します。

```
Rem ユニット番号 3 の登録状態をチェックします
If CheckUnit(3) = 1 Then
   Rem 処理ユニットのタイトル名を取得します
   TITLE$ = UnitTitle$(3)
Endif
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「AssignUnit (P.8-32)」
「CopyUnit (P.8-54)」
「DeleteUnit (P.8-66)」
「InsertUnit (P.8-152)」
「MeasureStart (P.8-182)」
「MeasureStop (P.8-183)」
「MoveUnit (P.8-190)」
「UnitCount (P.8-310)」

Chr$

指定したキャラクタコードに対応する文字を求めます。

・書式

Chr$(<expression>)

・パラメータ

パラメータ名	データ型	説明
<expression>	整数型	文字を求める数式（0～255）

・戻り値

文字列型の文字を返します。

・解説

<expression> パラメータで指定した ASCII コードのキャラクタコードに対応する文字を求めます。
<expression> パラメータにはコントロールコードを指定することもできます。
Chr$ 関数の逆関数には Asc 関数があります。Asc 関数は、指定したキャラクタ文字に対応するキャラクタコードを 10 進数で返します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

数値「48」と「13」のキャラクタ文字を求めます。

```
CHARA1$ = Chr$(48)
CHARA2$ = Chr$(13)
```

結果は以下のとおりになります。

```
CHARA1$ = "0"
CHARA2$ = "CR"
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「Asc (P.8-31)」
「Hex$ (P.8-142)」
「LCase$ (P.8-162)」
「Left$ (P.8-163)」
「Len (P.8-164)」
「Mid$ (P.8-186)」
「Piece$ (P.8-200)」
「Right$ (P.8-223)」
「Str$ (P.8-287)」
「Str2$ (P.8-288)」
「UCase$ (P.8-309)」
「Val (P.8-321)」

ClearMeasureData

処理ユニットの計測結果をクリアします。

・書式

ClearMeasureData [<unitNo>]

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	処理ユニット番号（0～（現在シーンの処理ユニット数 -1)）

・戻り値

なし

・解説

<unitNo> パラメータで指定した処理ユニットの計測結果をクリアします。
シーン制御マクロ／通信コマンドマクロでは、<unitNo> パラメータを省略した場合または-1 を指定した場合は、すべての処理ユニットの計測結果をクリアします。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。(詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。)
- ユニットマクロでは<unitNo>を省略することはできません。

・使用例

通信コマンドマクロで、処理ユニット番号 2 のサーチ処理ユニットの計測結果をクリアします。

```
Rem 計測結果をクリアします
ClearMeasureData 2
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「UnitNo (P.8-318)」

ClearScene

シーンデータをクリアします。

・書式

ClearScene <sceneNo>

・パラメータ

パラメータ名	データ型	説明
<sceneNo>	整数型	クリアするシーンのシーン番号（0～127）

・戻り値

なし

・解説

<sceneNo> パラメータで指定したシーン番号のシーンの設定内容をクリアして、工場出荷時の状態に戻します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定しても、Illegal function call のエラーは発生しません。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）

・使用例

シーン番号 2 のシーンをクリアします。

```
ClearScene 2
```

・使用可能なモジュール

通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「ChangeSceneGroup (P.8-39)」
「MeasureStart (P.8-182)」
「MeasureStop (P.8-183)」
「SceneCount (P.8-235)」
「SceneGroupNo (P.8-238)」
「SceneNo (P.8-241)」

ClearSceneGroup

シーングループデータをクリアします。

・書式

ClearSceneGroup <sceneGroupNo>

・パラメータ

パラメータ名	データ型	説明
<sceneGroupNo>	整数型	クリアするシーングループのシーングループ番号 0 ～ 31：シーングループ番号（0 ～ 31） -1：現在のシーングループ番号

・戻り値

なし

・解説

<sceneGroupNo> パラメータで指定したシーングループ番号のシーングループの設定内容と保存データを削除します。
<sceneGroupNo> パラメータに-1 を指定した場合は、現在のシーングループの設定内容と保存データを削除します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）

・使用例

シーングループ番号 1 のシーングループデータをクリアします。

```
ClearSceneGroup 1
```

・使用可能なモジュール

通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「ClearScene (P.8-43)」
「MeasureStart (P.8-182)」
「MeasureStop (P.8-183)」
「SceneGroupCount (P.8-237)」
「SceneGroupNo (P.8-238)」

Close

ファイルを閉じます。

・書式

Close [#<fileNo>[, #<fileNo>]...]

・パラメータ

パラメータ名	データ型	説明
<fileNo>	整数型	閉じるファイルのファイル番号（0～15）

・戻り値

なし

・解説

<fileNo> パラメータで指定したファイル番号のファイルを閉じます。
<fileNo> パラメータには、Open 関数でファイルを開いたときに指定したファイル番号を指定してください。
<fileNo> パラメータで複数のファイル番号を指定した場合は、開いている複数のファイルを閉じます。
<fileNo> パラメータを省略した場合は、開いているすべてのファイルを閉じます。
<fileNo> パラメータとして、0～15 の範囲を超える値を指定した場合は、Illegal function call のエラーが発生します。
Open 関数を実行して開いたファイルは、必ずこのマクロ関数を実行して同じ処理の中で閉じてください。このマクロ関数を実行しない場合や Open 関数を実行したサブルーチンとは別のサブルーチンや実行タイミングで実行した場合は、ファイルへの書き込みやファイルからの読み込みなどのファイルへのアクセスが正しく終了しない可能性があります。
このマクロ関数を実行して閉じたファイルにアクセスする場合は、もう一度 Open 関数を実行してファイルを開いてください。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

ファイルを開いてデータを書き込んだ後、ファイルを閉じます。

```
DATA& = 10

Rem ファイルを開きます
Open "E:¥input.dat" For Output As #1

Rem 開いたファイルにデータを書き込みます
Print #1 DATA&

Rem 開いたファイルを閉じます
Close #1
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「Open For Append As# (P.8-192)」
「Open For Input As# (P.8-193)」
「Open For Output As# (P.8-194)」
「Print (P.8-202)」

CloseTextData

メッセージファイルを閉じます。

・書式

CloseTextData [#<textDataNo>[, #<textDataNo>]...]

・パラメータ

パラメータ名	データ型	説明
<textDataNo>	整数型	閉じるメッセージファイルのテキストデータ番号（0～15）

・戻り値

なし

・解説

<textDataNo> パラメータで指定したテキストデータ番号のメッセージファイルを閉じます。
<textDataNo> パラメータとして、OpenTextData 関数でメッセージファイルを開いたときに指定したテキストデータ番号を指定してください。
<textDataNo> パラメータで複数のテキストデータ番号を指定した場合は、開いている複数のメッセージファイルを閉じます。
<textDataNo> パラメータを省略した場合は、開いているすべてのメッセージファイルを閉じます。
<textDataNo> パラメータとして、0～15 の範囲を超える値を指定した場合は、Illegal function call のエラーが発生します。
OpenTextData 関数を実行して開いたメッセージファイルは、必ずこのマクロ関数を実行して同じ処理の中で閉じてください。このマクロ関数を実行しない場合や、OpenTextData 関数を実行したサブルーチンとは別のサブルーチンや実行タイミングで実行した場合は、メッセージファイルを正しく閉じられず、その後の処理で正しくこのマクロ関数を実行できない可能性があります。
このマクロ関数を実行して閉じたメッセージファイルにアクセスする場合は、もう一度 OpenTextData 関数を実行してメッセージファイルを開いてください。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

ユニットマクロ処理ユニットの*MEASUREDISPT サブルーチンの中で、処理ユニット番号 5 のサーチ処理ユニットで計測した相関値を、処理ユニットのメッセージファイルから取得した文字列とともにテキスト表示ウィンドウに表示します。相関値は外部参照データ番号 5 です。

```
*MEASUREDISPT

    Rem 計測結果を取得します
    GetUnitData 5, 5, CR#

    Rem メッセージファイルを開きます
    OpenTextData "Search" As #1

    Rem テキストを取得します
    TEXT$ = GetText$(#1, "Correlation")

    Rem テキスト表示ウィンドウにメッセージファイルから取得した文字列を改行なしで描画します
    DrawText TEXT$, UnitJudge(5), 0

    Rem テキスト表示ウィンドウに計測結果を描画します
    DrawText Str2$(CR#, 4, 4, 0, 0), UnitJudge(5), 1

    Rem メッセージファイルを閉じます
    CloseTextData

Return
```

結果は以下のとおりになります。

```
相関値：90.0000
```

・使用可能なモジュール

ユニット演算マクロ ／ ユニットマクロ

・対応バージョン

バージョン 5.00 以降

・関連項目

「DrawText (P.8-97)」
「GetUnitData (P.8-138)」
「UnitJudge (P.8-317)」
「GetText$ (P.8-134)」
「OpenTextData (P.8-196)」

Cls

コンソールウィンドウの表示をクリアします。

・書式

Cls

・パラメータ

なし

・戻り値

なし

・解説

システム状態監視コンソールウィンドウの表示をクリアします。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

システム状態監視コンソールウィンドウの表示をクリアします。

```
Cls
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 6.50 以降

・関連項目

「Cont (P.8-49)」
「Debug (P.8-63)」
「DebugPrint (P.8-64)」
「List (P.8-167)」
「Print (P.8-202)」
「SetStop (P.8-270)」
「SubList (P.8-292)」
「VarList (P.8-322)」

Cont

プログラムの実行を中断した状態からプログラムの実行を再開します。

・書式

Cont [<mode>]

・パラメータ

パラメータ名	データ型	説明
<mode>	整数型	プログラム実行再開方法 このパラメータは省略可能です。 パラメータを指定しない場合は、プログラムの実行が再開されプログラムが最後またはエラーが発生するまで実行されます。 0：ステップイン実行 現在のプログラム行がサブルーチンの呼び出しの場合は、サブルーチンの中に入ってステップ実行します。それ以外の場合は、現在のステートメントを実行して次のプログラム行で中断します。 1：ステップオーバー実行 現在のプログラム行がサブルーチンの呼び出しの場合は、サブルーチン全体を実行してサブルーチン呼び出し後の次のプログラム行で中断します。それ以外の場合は、現在のステートメントを実行して次のプログラム行で中断します。 2：ステップアウト実行 現在のプログラム行がサブルーチンから呼び出されたサブルーチンの場合は、現在のプログラム行以降のサブルーチン全体を実行してサブルーチン呼び出し元のサブルーチンの次のプログラム行で中断します。それ以外の場合は、プログラムが最後またはエラーが発生するまで実行されます。

・戻り値

なし

・解説

<mode> パラメータで指定したプログラムの実行再開方法で、Stop 関数でプログラムの実行を中断した状態からプログラムの実行を再開します。（詳細は、「6-1 デバッグ機能の使い方について (P.6-2)」を参照してください。）

<mode> パラメータを指定して実行することで、Stop 関数実行後にプログラムを 1 行ずつステップ実行することができます。

パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。

パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定しても、Illegal function call のエラーは発生しません。

マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

処理ユニット番号 1 のユニットマクロ処理ユニットのプログラム 220 行目に記述された Stop 関数が実行された後、次の 230 行目を 1 行だけステップ実行します。

```
Macro(U1) 220 Stop
Macro(U1) Stop in 220
Macro(U1) 230 POS.X#=(POS0.X@ + POS1.X@) / 2
Macro(U1)>Cont 1
Macro(U1)>
```

・使用可能なモジュール
ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 5.20 以降

・関連項目

「Cls (P.8-48)」
「DebugPrint (P.8-64)」
「Print (P.8-202)」
「SetVar (P.8-281)」
「SubList (P.8-292)」
「Debug (P.8-63)」
「List (P.8-167)」
「SetStop (P.8-270)」
「Stop (P.8-286)」
「VarList (P.8-322)」

ContinuousMeasure

連続計測の開始／停止を行います。

・書式
ContinuousMeasure <state>

・パラメータ

パラメータ名	データ型	説明
<state>	整数型	連続計測の実行状態 False：連続計測停止 True：連続計測開始

・戻り値
なし

・解説
<state> パラメータで、連続計測の開始／停止を指定します。
引数として True を指定すると連続計測開始となり、False を指定すると連続計測停止となります。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。

・使用上の注意
このマクロ関数は、BUSY 信号などの計測中信号が OFF した計測許可状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）

・使用例

```
Rem 連続計測を開始、停止する
NO& = int(Argumentvalue#(0))

Rem 通信コマンドマクロで送られてきた値が 1 なら連続計測開始
If NO& = 1 then
   ContinuousMeasure true
   Rem 通信コマンドマクロで送られてきた値が 1 以外なら連続計測停止
Else
   ContinuousMeasure false
EndIf
```

・使用可能なモジュール
シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン

バージョン 5.40 以降

・関連項目

「If Then Else (P.8-143)」　　「MeasureStart (P.8-182)」

CopyMeasureImage

計測画像をユニットマクロ処理ユニットの画像としてコピーします。

・書式

CopyMeasureImage <measureImageNo>, <myImageNo>

・パラメータ

パラメータ名	データ型	説明
<measureImageNo>	整数型	計測画像番号（常に 0）
<myImageNo>	整数型	コピー先の画像番号（0～31）

・戻り値

なし

・解説

<measureImageNo> パラメータで指定した計測画像番号の画像を、<myImageNo> パラメータで指定した画像番号のこのマクロ関数を呼び出すユニットマクロ処理ユニットの画像バッファにコピーします。
<measureImageNo> パラメータには、通常は 0 を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

計測ごとに計測画像を順に画像バッファ上に蓄積します。

```
Rem 画像バッファ番号を算出します
MYIMAGENO& = MYIMAGENO& + 1

Rem 32 枚目以降は古い順に上書きします
If MYIMAGENO& > 31 Then
   MYIMAGENO& = 0
Endif

Rem 計測画像をユニットマクロ処理ユニットの画像バッファに蓄積します
CopyMeasureImage 0, MYIMAGENO&
```

・使用可能なモジュール

ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「CopyUnitImage (P.8-56)」　「SetMeasureImage (P.8-256)」

CopyScene

シーンデータをコピーします。

・書式

CopyScene <srcSceneNo>, <destSceneNo>

・パラメータ

パラメータ名	データ型	説明
<srcSceneNo>	整数型	コピー元のシーンのシーン番号（0～127）
<destSceneNo>	整数型	コピー先のシーンのシーン番号（0～127）

・戻り値

なし

・解説

<srcSceneNo> パラメータで指定したシーン番号のシーンデータを、<destSceneNo> パラメータで指定したシーン番号のシーンデータにコピーします。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
<srcSceneNo> パラメータと<destSceneNo> パラメータに同じシーン番号を指定した場合は、Illegal function call のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
データをコピーするために必要なワークメモリの空き容量が不足している場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）

・使用例

シーン 2 の内容をシーン 3 にコピーします。

```
CopyScene 2, 3
```

・使用可能なモジュール

通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「CopySceneGroup (P.8-53)」　「MeasureStart (P.8-182)」
「MeasureStop (P.8-183)」　「SceneCount (P.8-235)」
「SceneGroupNo (P.8-238)」　「SceneNo (P.8-241)」

CopySceneGroup

シーングループデータをコピーします。

・書式

CopySceneGroup <srcSceneGroupNo>, <destSceneGroupNo>

・パラメータ

パラメータ名	データ型	説明
<srcSceneGroupNo>	整数型	コピー元のシーングループのシーングループ番号（0～31）
<destSceneGroupNo>	整数型	コピー先のシーングループのシーングループ番号（0～31）

・戻り値

なし

・解説

<srcSceneGroupNo> パラメータで指定したシーングループ番号のシーングループデータを、<destSceneGroupNo> パラメータで指定したシーングループ番号のシーングループデータにコピーします。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
<srcSceneGroupNo> パラメータと<destSceneGroupNo> パラメータに同じシーングループ番号を指定した場合は、Illegal function call のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
データをコピーするために必要なワークメモリの空き容量が不足している場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）

・使用例

シーングループ 0 の内容をシーングループ 1 にコピーします。

```
CopySceneGroup 0, 1
```

・使用可能なモジュール

通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「CopyScene (P.8-52)」
「MeasureStart (P.8-182)」
「MeasureStop (P.8-183)」
「SceneCount (P.8-235)」
「SceneGroupNo (P.8-238)」
「SceneNo (P.8-241)」

CopyUnit

処理ユニットをコピーします。

・書式

CopyUnit [<srcSceneNo>,] <srcUnitNo>, <destUnitNo>, <mode>

・パラメータ

パラメータ名	データ型	説明
<srcSceneNo>	整数型	コピー元のシーン番号（0～127）
<srcUnitNo>	整数型	コピー元の処理ユニット番号（0～（コピー元のシーンの処理ユニット数 -1)）
<destUnitNo>	整数型	コピー先の処理ユニット番号（0～（現在シーンの処理ユニット数)）
<mode>	整数型	処理モード 0：コピー先の処理ユニット番号の処理ユニットに上書き 1：コピー先の処理ユニット番号の位置に挿入

・戻り値

なし

・解説

<srcSceneNo> パラメータで指定したシーン番号の <srcUnitNo> パラメータで指定した処理ユニットを、現在のシーンの<destUnitNo> パラメータで指定した計測フロー上の位置に、<mode> パラメータで指定した処理モードでコピーします。
<srcSceneNo> パラメータが省略された場合、コピー元のシーンは現在のシーンとなります。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
コピー元の処理ユニットに未切替のシーンのカメラ画像入力（処理ユニット 0）を指定すると illlegal function call のエラーが発生します。
未切替のシーンには処理ユニットが入っておらず、シーン切替後にカメラ画像入力が処理ユニット 0 として入力されます。CopyUnit コマンドを実行する前にシーン切替を行ってください。
<mode> パラメータで 0 を指定した場合は、コピー先の処理ユニット番号の処理ユニットがコピー元の処理ユニットに上書きされます。<mode> パラメータで 1 を指定した場合は、コピー先の処理ユニット番号にコピー元の処理ユニットが挿入されます。このときコピー先の処理ユニットの処理ユニット番号は 1 繰り下げられます。
<srcSceneNo> パラメータを省略し、<mode> パラメータで 0 を指定した場合は、<srcUnitNo> パラメータと<destUnitNo> パラメータに異なる値を指定してください。同じ値を指定した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）

・使用例

シーン 2 の処理ユニット 3 を現在のシーンの処理ユニット 4 の前に挿入コピーします。

```
CopyUnit 2, 3, 4, 1
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「AssignUnit (P.8-32)」
「CheckUnit (P.8-40)」
「DeleteUnit (P.8-66)」
「InsertUnit (P.8-152)」
「MeasureStart (P.8-182)」
「MeasureStop (P.8-183)」
「MoveUnit (P.8-190)」
「UnitCount (P.8-310)」

CopyUnitFigure

処理ユニットの図形データをコピーします。

・書式
CopyUnitFigure <srcSceneNo>, <srcUnitNo>, <srcFigureNo>, <destUnitNo>, <destFigureNo>

・パラメータ

パラメータ名	データ型	説明
<srcSceneNo>	整数型	コピー元のシーン番号（0～127）
<srcUnitNo>	整数型	コピー元のユニット番号（0～（コピー元のシーンの処理ユニット数 -1））
<srcFigureNo>	整数型	コピー元の図形番号（参照：「9-7 図形番号一覧 (P.9-70)」）
<destUnitNo>	整数型	コピー先のユニット番号（0～（現在シーンの処理ユニット数 -1））
<destFigureNo>	整数型	コピー先の図形番号（参照：「9-7 図形番号一覧 (P.9-70)」）

・戻り値
なし

・解説
<srcSceneNo> パラメータで指定したシーンの、<srcUnitNo> パラメータで指定した処理ユニットの、<srcFigureNo> パラメータで指定した図形番号の図形データを、<destUnitNo> パラメータで指定した処理ユニットの、<destFigureNo> パラメータで指定した図形番号の図形データにコピーします。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）
- このマクロ関数は、1 回以上計測するか画像ファイルを指定して再計測した後、計測画像が表示された状態で使用してください。
- 設定する図形に計測画像外の画素が含まれないように図形データを設定してください。

・使用例
シーン 0 計測フローにある処理ユニット番号 2 の形状サーチ III 処理ユニットの計測領域図形を、処理ユニット番号 5 の形状サーチ III 処理ユニットの計測領域にコピーします。形状サーチ III 処理項目の計測領域図形は図形番号 1 です。

```
Rem 計測領域の図形をコピーします
CopyUnitFigure 0, 2, 1, 5, 1
```

・使用可能なモジュール
シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「GetUnitFigure (P.8-139)」
「MeasureStart (P.8-182)」
「MeasureStop (P.8-183)」
「SceneNo (P.8-241)」
「SetUnitFigure (P.8-276)」
「UnitNo (P.8-318)」
「Ut (P.8-320)」

CopyUnitImage

処理ユニットの画像をユニットマクロ処理ユニットの画像としてコピーします。

・書式
CopyUnitImage <unitNo>, <imageNo>, <myImageNo>

・パラメータ

パラメータ名	データ型	説明
<UnitNo>	整数型	コピー元の処理ユニット番号（0～（現在シーンの処理ユニット数-1)）
<imageNo>	整数型	コピー元の画像番号（参照：「9-9 画像番号一覧 (P.9-74)」）
<myImageNo>	整数型	コピー先の画像番号（0～31）

・戻り値
なし

・解説
<unitNo> パラメータで指定した処理ユニットの、<imageNo> パラメータで指定した画像番号の画像を、<myImageNo> パラメータで指定した画像番号のこのマクロ関数を呼び出すユニットマクロ処理ユニットの画像バッファにコピーします。
<imageNo> パラメータには、通常は 0 を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）

・使用例
処理ユニット番号 1 のカラーグレーフィルタ処理ユニットの変換後画像を、計測ごとに順に画像バッファ上に蓄積します。

```
Rem 画像バッファ番号を算出します
MYIMAGENO& = MYIMAGENO& + 1

Rem 32 枚目以降は古い順に上書きします
If MYIMAGENO& > 31 Then
   MYIMAGENO& = 0
Endif

Rem 変換後画像をユニットマクロ処理ユニットの画像バッファに蓄積します
CopyUnitImage 1, 0, MYIMAGENO&
```

・使用可能なモジュール
ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「CopyMeasureImage (P.8-51)」
「SetMeasureImage (P.8-256)」
「UnitNo (P.8-318)」
「Ut (P.8-320)」

CopyUnitModel

処理ユニットのモデルデータをコピーします。

・書式
CopyUnitModel <srcSceneNo>, <srcUnitNo>, <srcModelNo>, <destUnitNo>, <destModelNo>

・パラメータ

パラメータ名	データ型	説明
<srcSceneNo>	整数型	コピー元のシーン番号（0～127）
<srcUnitNo>	整数型	コピー元のユニット番号（0～（コピー元のシーンの処理ユニット数 -1））
<srcModelNo>	整数型	コピー元のモデル番号（参照：「9-8 モデル番号一覧 (P.9-72)」）
<destUnitNo>	整数型	コピー先のユニット番号（0～（現在シーンの処理ユニット数 -1））
<destModelNo>	整数型	コピー先のモデル番号（参照：「9-8 モデル番号一覧 (P.9-72)」）

・戻り値
なし

・解説
<srcSceneNo> パラメータで指定したシーンの、<srcUnitNo> パラメータで指定した処理ユニットの、<srcModelNo> パラメータで指定したモデル番号のモデルデータを、<destUnitNo> パラメータで指定した処理ユニットの、<destModelNo> パラメータで指定したモデル番号のモデルデータにコピーします。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。
処理項目が計測に使用するモデルには、モデルデータ、モデル図形、モデルパラメータなどがあります。モデルの内容は処理項目によって異なり、モデルデータをコピーするだけでは正しく動作しない

場合がありますが、シーン制御マクロの中でバリアント型変数を使用してモデルデータをコピーできるものがあります。詳しくは、Application Producer（形 FH-AP1、別売）を参照してください。

・使用上の注意

- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）
- このマクロ関数は、1 回以上計測するか画像ファイルを指定して再計測した後、計測画像が表示された状態で使用してください。
- 設定する図形に計測画像外の画素が含まれないように図形データを設定してください。

・使用例

処理ユニット番号 2 の形状サーチⅢ処理ユニットのモデルを、処理ユニット番号 3 の形状サーチⅢ処理ユニットにコピーします。

```
Rem 形状サーチⅢのモデル図形をコピーします
CopyUnitFigure 0, 2, 0, 3, 0

Rem 形状サーチⅢの検出点の設定をコピーします
GetUnitData 2, "detectionPosX", PosX#
GetUnitData 2, "detectionPosY", PosY#
SetUnitData 3, "detectionPosX", PosX#
SetUnitData 3, "detectionPosY", PosY#

Rem 形状サーチⅢのモデルデータをコピーします
CopyUnitModel 0, 2, 0, 3, 0
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「CopyUnitFigure (P.8-55)」
「GetUnitData (P.8-138)」
「MeasureStart (P.8-182)」
「MeasureStop (P.8-183)」
「SceneNo (P.8-241)」
「SetUnitData (P.8-275)」
「UnitNo (P.8-318)」
「Ut (P.8-320)」

Cos

指定した数式の余弦（コサイン）を求めます。

・書式

Cos(<expression>)

・パラメータ

パラメータ名	データ型	説明
<expression>	整数型 実数型	余弦を求める数式

・戻り値

実数型の余弦の値を-1 から 1 までの範囲の値で返します。

・解説

<expression> パラメータで指定した数式の余弦を求めます。
<expression> パラメータには値をラジアン値で指定します。角度値をラジアン値に変換するには、角度値に π/180 をかけます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

60° の余弦を求めます。

```
DATA# = Cos(60/180*3.141592)
```

結果は以下のとおりになります。

```
DATA# = 0.5
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「Atn (P.8-33)」
「Sin (P.8-283)」
「UnitData (P.8-311)」
「GetUnitData (P.8-138)」
「Tan (P.8-295)」

Crspoint

2 直線の交点を求めます。

・書式

Crspoint <line1()>, <line2()>, <x>, <y>

・パラメータ

パラメータ名	データ型	説明
<line1()>	実数型配列	交点を求める直線 1 の直線成分
<line2()>	実数型配列	交点を求める直線 2 の直線成分
<x>	実数型	求めた交点の X 座標格納領域
<y>	実数型	求めた交点の Y 座標格納領域

・戻り値

なし

・解説

<line1()> パラメータで指定した直線成分を持つ直線と、<line2()> パラメータで指定した直線成分を持つ直線の交点を求めます。<x> パラメータと<y> パラメータには、求めた交点の X 座標と Y 座標をそれぞれ格納する変数を指定してください。

<line1()> パラメータと<line2()> パラメータには、直線の方程式 ax+by+c=0 を満たす a,b,c のパラメータについて、配列要素 0 に a、1 に b、2 に c をそれぞれ格納した 1 次元の実数型配列変数を、要素番号を指定せず() だけ付加して指定してください。

このマクロ関数は、おもに Lsqumeth 関数で求めた直線の交点を求めるときに使用します。

パラメータとして、定義されていない配列を指定した場合は、Undefined array のエラーが発生します。

パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。

パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。

マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

求めた 2 直線の交点座標を求めます。2 直線はそれぞれ、処理ユニット 1～4 および処理ユニット 5～8 を用いて求めています。

```
Dim POS1X#(3), POS1Y#(3), POS2X#(3), POS2Y#(3), PARAM1#(2), PARAM2#(2)

Rem 直線 1 の値を初期化します
For I&=0 To 3
   GetUnitData I&+1, "X", POS1X#(I&)
   GetUnitData I&+1, "Y", POS1Y#(I&)
Next

Rem 直線 1 成分を求めます
Lsqumeth 4, POS1X#(), POS1Y#(), PARAM1#()

Rem 直線 2 の値を初期化します
For I&=0 To 3
   GetUnitData I&+5, "X", POS2X#(I&)
   GetUnitData I&+5, "Y", POS2Y#(I&)
Next

Rem 直線 2 成分を求めます
Lsqumeth 4, POS2X#(), POS2Y#(), PARAM2#()

Rem 2 直線の交点を求めます
Crspoint PARAM1#(), PARAM2#(), CRSX#, CRSY#
Erase POS1X#(), POS1Y#(), POS2X#(), POS2Y#(), PARAM1#(), PARAM2#()
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「Erase (P.8-105)」
「Lsqumeth (P.8-175)」
「GetUnitData (P.8-138)」
「UnitData (P.8-311)」

Date$

内蔵クロックの日付を読み出します。

・書式
Date$

・パラメータ
なし

・戻り値
文字列型の日付の値を返します。
日付の値は、内蔵クロックの日付を年（YY）、月（MM）、日（DD）それぞれスラッシュ（/）で区切った文字列です。年月日それぞれの範囲は以下のとおりです。
- 年（YY）：00～80
- 月（MM）：01～12
- 日（DD）：01～31

・解説
内蔵クロックの日付を年月日の文字列として読み出します。
年の値は 2000 年から 2080 年を、00 から 80 として表現されます。
内蔵クロックの設定はシステム設定の日時設定で設定します。（詳細は、『画像処理システム FH/FHV シリーズ ユーザーズマニュアル(SDNB-719)』の「日付と時刻を合わせる[その他]」を参照してください。）
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
内蔵クロックの日付を読み出して、システム状態監視コンソールウィンドウに出力します。

```
Rem 内蔵クロックの日付を読み出します
TODAY$ = Date$

Rem 読み出した日付をシステム状態監視コンソールウィンドウに出力します
Print "20";TODAY$
```

結果は以下のとおりになります。

```
2011/03/10
```

・使用可能なモジュール
ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「GetSystemData (P.8-133)」
「Piece$ (P.8-200)」
「SetSystemData (P.8-271)」
「Mid$ (P.8-186)」
「Print (P.8-202)」
「Time$ (P.8-297)」

Debug

プログラムの実行形態と情報出力方法を設定します。

・書式

Debug <mode>

・パラメータ

パラメータ名	データ型	説明
<mode>	整数型	実行形態と情報出力方法 0：リリースモード、エラー発生時にエラー内容出力なし 1：リリースモード、エラー発生時にエラー内容をシステム状態監視コンソールウィンドウに出力 2：リリースモード、プログラム実行時に各行内容をシステム状態監視コンソールウィンドウに出力 3：リリースモード、エラー発生時にエラー内容をメッセージボックスに出力 16： デバッグモード、エラー発生時にエラー内容出力なし 17： デバッグモード、エラー発生時にエラー内容をシステム状態監視コンソールウィンドウに出力 18：デバッグモード、プログラム実行時に各行内容をシステム状態監視コンソールウィンドウに出力 19：デバッグモード、エラー発生時にエラー内容メッセージボックス出力

・戻り値

なし

・解説

<mode> パラメータで指定したプログラムの実行形態と情報出力方法を設定します。
デバッグモードを設定することで、デバッグモード時のみ実行されるマクロ関数を使用してプログラムをデバッグすることができます。デバッグ終了後はリリースモードに設定することで、DebugPrint 関数などのデバッグモード時のみ実行されるマクロ関数をプログラムから削除しなくて済みます。（詳細は、「6-1 デバッグ機能の使い方について (P.6-2)」を参照してください。）
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

ユニットマクロ処理ユニットの MCRINIT サブルーチンで、プログラムの実行形態をデバッグモード、エラー発生時にエラー内容をシステム状態監視コンソールウィンドウに出力するよう設定します。

```
*MCRINIT

   Rem デバッグモードでエラー発生時にエラーをシステム状態監視コンソールウィンドウに出力するよう設定
します
   Debug 17

Return
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 5.20 以降

・関連項目

「Cls (P.8-48)」
「Cont (P.8-49)」
「DebugPrint (P.8-64)」
「List (P.8-167)」
「Print (P.8-202)」
「SetStop (P.8-270)」
「SetVar (P.8-281)」
「Stop (P.8-286)」
「SubList (P.8-292)」
「VarList (P.8-322)」

DebugPrint

システム状態監視コンソールウィンドウにデバッグ情報を出力します。

・書式
DebugPrint <expression>[;¦, <expression>...]

・パラメータ

パラメータ名	データ型	説明
<expression>	整数型 実数型 文字列型 配列型	出力する数式や文字列

・戻り値
なし

・解説
<expression> パラメータで指定した数式や文字列を、システム状態監視コンソールウィンドウに出力します。(詳細は、「3-1-2 システム状態監視コンソールウィンドウの見かた (P.3-6)」を参照してください。)
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
- このマクロ関数は、Debug 関数でデバッグモードを指定している場合にのみ有効です。それ以外の場合は、Rem 関数と同様に、プログラム行の本関数の記述は無視されます。(詳細は、「6-1 デバッグ機能の使い方について (P.6-2)」を参照してください。
- システム状態監視コンソールウィンドウにデータを出力すると、システム状態監視コンソールウィンドウがセンサコントローラのメイン画面の前面に表示されます。メイン画面を前面に表示する場合は、システム状態監視コンソールウィンドウ右上の[_]をクリックするか、センサコントローラに USB キーボードを接続して、[Alt] + [Tab] で表示する画面を切り替えてください。

・使用例
デバッグモード時に、システム状態監視コンソールウィンドウにデバッグ文字列を出力します。

```
Rem 実行形態をデバッグモードに設定します
Debug 18

Rem 文字列「Result = OK」をデバッグ情報として出力します
DebugPrint "Result = " + "OK"

Rem 実行形態をリリースモードに設定します
Debug 1
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 5.20 以降

・関連項目

「Cls (P.8-48)」
「Cont (P.8-49)」
「Debug (P.8-63)」
「List (P.8-167)」
「Print (P.8-202)」
「SetStop (P.8-270)」
「SetVar (P.8-281)」
「Stop (P.8-286)」
「SubList (P.8-292)」
「VarList (P.8-322)」

DeleteProfileData

設定ファイル（.ini ファイル）のデータを削除します。

・書式

DeleteProfileData <fileName>, <section>[, <deleteSectionHeader> ¦ <key>]

・パラメータ

パラメータ名	データ型	説明
<fileName>	文字列型	対象ファイルの絶対パス名
<section>	文字列型	対象のセクション名
<deleteSectionHeader>	整数型	セクション削除の有無 • 0：セクションを削除しない • 1：セクションを削除する
<key>	文字列型	対象のキー名

・戻り値

なし

・解説

<fileName>パラメータで指定したファイルの<section>パラメータで指定したセクション内のデータを削除します。
<deleteSectionHeader>パラメータに 1 以上の整数を入れた場合、<section>パラメータで指定したセクションごと削除します。
<key>パラメータを指定した場合、<section>パラメータで指定したセクションに含まれる該当のキー名のデータのみが削除されます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例
.ini ファイルの Setting セクション内の、キー名 Logging のデータを削除します。

```
DeleteProfileData "E:¥Setting.ini", "Setting", "Logging"
```

・使用可能なモジュール
シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 6.40 以降

・関連項目

「GetProfileData (P.8-130)」　「SetProfileData (P.8-262)」

DeleteUnit

処理ユニットを削除します。

・書式
DeleteUnit <unitNo>

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	削除する処理ユニットの処理ユニット番号（0～（現在シーンの処理ユニット数 -1)）

・戻り値
なし

・解説
<unitNo> パラメータで指定した現在のシーンの処理ユニットを、計測フローから削除します。
<unitNo> パラメータで指定した処理ユニット番号以降の処理ユニットは、処理ユニット番号が 1 繰り上げられます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）

・使用例
ユニット番号 2 の処理ユニットを削除します。

```
DeleteUnit 2
```

・使用可能なモジュール
シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「AssignUnit (P.8-32)」
「CopyUnit (P.8-54)」
「MeasureStart (P.8-182)」
「MoveUnit (P.8-190)」
「CheckUnit (P.8-40)」
「InsertUnit (P.8-152)」
「MeasureStop (P.8-183)」
「UnitCount (P.8-310)」

Dim

配列変数を定義します。

・書式

Dim <arrayName>(<maxCount>[, <maxCount>[, <maxCount>[, <maxCount>]]])

・パラメータ

パラメータ名	データ型	説明
<arrayName>	-	使用する配列の変数名
<maxCount>	整数型	添字の最大値

・戻り値

なし

・解説

<arrayName> パラメータで指定した変数名の、<maxCount> パラメータで指定した添字の最大値を持つ 1 次元から 4 次元までの配列変数を定義します。
<arrayName> パラメータの末尾には、配列変数のデータ型を意味する型識別子を指定してください。（詳細は、「4-1-3 変数 (P.4-6)」を参照してください。）
このマクロ関数で定義した配列変数は、Erase 関数を実行して解放してください。
次元が異なる配列変数は、同じ変数名であれば同じ変数として扱われます。
配列変数と変数は、同じ変数名であっても異なる変数として扱われます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。
Option Explicit コマンドによるマクロ変数チェックを行うために、一時変数、配列変数、参照変数の定義を行うために使用します。（詳細は、「マクロ変数チェック機能 (P.4-9)」を参照してください。）

・使用上の注意

- Option Explicit コマンドの使用によって振る舞いが変わります。
  Option Explicit コマンド使用時、コマンドが変数の未定義や重複をチェックするために、すでに定義している配列変数を再定義した場合は、重複定義のエラーが発生します。
  Option Explicit コマンド不使用時、すでに定義している配列変数を再定義した場合は、以前に定義した配列変数を解放してから再定義します。

・使用例

配列を定義します。

```
Dim XY&(3)
Dim XY#(7, 15)
Dim CHARA$(31, 63, 127, 255)
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 5.40 以降

・関連項目

「Erase (P.8-105)」
「Option Explicit (P.8-198)」
「ReDim (P.8-214)」

DisableUnit

指定ユニットの計測 ON/OFF を切り替えます。

・書式

DisableUnit <unitNo>, <disabled>

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	処理ユニット番号
<disabled>	整数型	0：計測を ON にする 1：計測を OFF にする

・戻り値

なし

・解説

<unitNo> パラメータで指定した処理ユニットの計測 ON/OFF 状態を<disabled> プロパティで指定された状態に変更します。
<unitNo> パラメータに存在しないユニット番号を指定すると Illegal function call エラーが発生します。
<disabled> パラメータに未定義の値を入れてもエラーは発生しません。
フロー先頭のユニットを計測 OFF 状態にすることはできません。Illegal function call エラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）

・使用例

ユニット番号 3 のユニットを計測 OFF 状態にします。

```
MeasureStop
DisableUnit 3, 1
MeasureStart
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン

バージョン 6.50 以降

・関連項目

「MeasureStart (P.8-182)」
「MeasureStop (P.8-183)」

DisplaySubNo

画像表示サブ番号を取得します。

・書式
DisplaySubNo

・パラメータ
なし

・戻り値
整数型の画像表示サブ番号の値を返します。

・解説
メイン画面の画像表示ウィンドウに設定されている画像表示サブ番号を取得します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
ユニットマクロ処理ユニットの*MEASUREDISPG サブルーチンの中で、メイン画面の画像表示ウィンドウに設定されている画像表示サブ番号によって、画像表示ウィンドウにグラフィック表示する内容を変えます。

```
*MEASUREDISPG

   Rem 表示サブ番号を取得します
   SUBNO& = DisplaySubNo

   Rem 表示サブ番号によって画像表示ウィンドウに表示する内容を変えます
   Select SUBNO&
   Case 1
      Rem 表示サブ番号が 1 のときは処理ユニット 1 のタイトルを判定結果に応じた色で表示します
      SetTextStyle 24, TA_LEFT, UnitJudge(1), 0, FONTSTYLE_NORMAL
      TEXT$ = UnitTitle$(1)
   Case 2
      Rem 表示サブ番号が 2 のときは処理ユニット 2 のタイトルを判定結果に応じた色で表示します
      SetTextStyle 24, TA_LEFT, UnitJudge(2), 0, FONTSTYLE_NORMAL
      TEXT$ = UnitTitle$(2)
   Case Else
      Rem 表示サブ番号が 1、2 以外のときは未計測色で表示します
      SetTextStyle 24, TA_LEFT, JUDGE_NC, 0, FONTSTYLE_NORMAL
      TEXT$ = "Error"
   End Select

   Rem 画像表示ウィンドウにテキストを表示します
   DrawTextG TEXT$, 50, 0, 0, UnitNo

Return
```

・使用可能なモジュール
ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「DrawTextG (P.8-99)」
「UnitJudge (P.8-317)」
「UnitTitle$ (P.8-319)」
「SetTextStyle (P.8-272)」
「UnitNo (P.8-318)」

DisplayUnitNo

フロー表示ウィンドウの処理ユニット番号の選択状態を取得します。

・書式
DisplayUnitNo

・パラメータ
なし

・戻り値
整数型の処理ユニット番号の値を返します。

・解説
フロー表示ウィンドウ上で選択状態のユニットの処理ユニット番号を取得します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
通信コマンドマクロにフロー表示ウィンドウで選択されている処理ユニット番号を取得します。

```
Rem フロー表示ウィンドウで選択されている処理ユニット番号を取得します
NO& = DisplayUnitNo

Rem 通信コマンドのレスポンスデータに処理ユニット番号を設定します
ResponseString$ = Str$(NO&)
CommandResponse& = 0
```

・使用可能なモジュール
シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「GetImageWindow (P.8-123)」
「UnitNo (P.8-318)」
「SetDisplayUnitNo (P.8-249)」
「Ut (P.8-320)」

Do Loop While

条件を満たしている間、Do から Loop までの間の文を繰り返し実行します。

・書式

Do
<statement>
Loop While <expression>

・パラメータ

パラメータ名	データ型	説明
<statement>	-	繰り返し実行するステートメント
<expression>	-	繰り返し実行の条件となる論理式（参照：「4-1-5 演算子 (P.4-12)」）

・戻り値

なし

・解説

<statement> パラメータで指定した Do ブロックのステートメントを実行して、<expression> パラメータで指定した論理式が真の場合にステートメントを繰り返し実行します。
Do ブロックのステートメントの中で、Exit Do ステートメントを使用した場合は、繰り返し実行を途中で強制的に終了します。
Do ブロックのステートメントの中で、Goto 関数や Gosub 関数を使用して Do ブロックの中から外に制御を移したり、外から中に制御を移したりした場合は、予期せぬ動作になる可能性があります。
Do 文と Loop While 文のいずれかを記述しなかった場合は、記述の内容によって DO without LOOP、LOOP without DO、EXIT without DO のいずれかのエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

ループカウンタが一定の値になるまで処理を繰り返します。

```
NUM& = 0

Rem 処理を繰り返します
Do
   NUM& = NUM& + 1
Loop While NUM& < 100

Print NUM&
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「For To Step Next (P.8-118)」　「Print (P.8-202)」

Dposline

指定した直線と点の最短距離を求めます。

・書式

Dposline (<x>, <y>, <line()>)

・パラメータ

パラメータ名	データ型	説明
<x>	実数型	距離を求める点の X 座標
<y>	実数型	距離を求める点の Y 座標
<line()>	実数型配列	距離を求める直線の直線成分

・戻り値

実数型の最短距離の値を返します。

・解説

<x> パラメータで指定した X 座標と、<y> パラメータで指定した Y 座標を持つ点と、<line()> パラメータで指定した直線成分を持つ直線の最短距離を求めます。
<line()> パラメータには、直線の方程式 ax+by+c=0 を満たす a,b,c のパラメータについて、配列要素 0 に a、1 に b、2 に c をそれぞれ格納した 1 次元の実数型配列変数を、要素番号を指定せず() だけ付加して指定してください。
このマクロ関数は、おもに Lsqumeth 関数で求めた直線に対して、元になった点の分散や偏差などを求めるときに使用します。
パラメータとして、定義されていない配列を指定した場合は、Undefined array のエラーが発生します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

4 つの点から求めた直線に対して、その分散と偏差を求めます。

```
Dim POSX#(3), POSY#(3), PARAM#(2), DIST#(3)

Rem 直線の値を初期化します
For I&=0 To 3
   GetUnitData I&+1, "X", POSX#(I&)
   GetUnitData I&+1, "Y", POSY#(I&)
Next

Rem 直線成分を求めます
Lsqumeth 4, POSX#(), POSY#(), PARAM#()
SUMDIST# = 0
For I&=0 To 3
   Rem 直線と点の最短距離を求めます
   DIST#(I&) = Dposline(POSX#(I&), POSY#(I&), PARAM#())
   SUMDIST# = SUMDIST# + DIST#(I&)
Next

Erase POSX#(), POSY#(), PARAM#(), DIST#()
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「Erase (P.8-105)」
「Lsqumeth (P.8-175)」
「GetUnitData (P.8-138)」
「UnitData (P.8-311)」

DrawArc

画像表示ウィンドウに円弧を描画します。

・書式

DrawArc <x>, <y>, <radius>, <start>, <end>, <imageNo>[, <unitNo>]

・パラメータ

パラメータ名	データ型	説明
<x>	整数型	描画する円弧の中心 X 座標
<y>	整数型	描画する円弧の中心 Y 座標
<radius>	整数型	描画する円弧の半径
<start>	整数型	描画する円弧の開始点角度
<end>	整数型	描画する円弧の終了点角度
<imageNo>	整数型	描画対象の計測画像の番号（常に 0）
<unitNo>	整数型	描画を実行する処理ユニットの処理ユニット番号（0～（現在シーンの処理ユニット数 -1））

・戻り値

なし

・解説

<x> パラメータと<y> パラメータで指定した点を中心とする、<radius> パラメータで指定した半径の円の、<start> パラメータで指定した開始点角度から<end> パラメータで指定した終了点角度までの円弧を、<imageNo> パラメータで指定した画像番号の計測画像に対してグラフィック描画します。
位置ずれ修正前の座標位置にグラフィックを描画したい場合は、<unitNo> パラメータに当該ユニットマクロの処理ユニット番号を指定してください。<unitNo> パラメータを省略した場合は、位置ずれ修正後の座標位置にグラフィックが描画されます。
<x> パラメータと<y> パラメータには、画像左上を原点とするカメラ座標を指定してください。
<start> パラメータと<end> パラメータには、カメラ座標 X 軸正方向を 0 度とする角度値を、右回りを正方向の角度として指定してください。
<imageNo> パラメータには、通常は 0 を指定してください。
<unitNo> パラメータには、通常は当該処理を実行するユニットマクロ処理ユニットの処理ユニット番号を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、*MEASUREDISPI サブルーチンまたは*MEASUREDISPG サブルーチンの中でしか使用できません。それ以外のサブルーチンで使用しようとした場合は、Illegal function call のエラーが発生します。
- 描画する前に、SetDrawStyle で線の種類や色や幅を設定してください。設定しなければ、直前の設定で描画されます。

・使用例

ユニットマクロ処理ユニットの*MEASUREDISPG サブルーチンの中で、処理ユニット番号 5 の円形スキャンエッジ位置処理ユニットで計測した位置に、計測した半径で、開始点角度-90 度から終了点

角度 180 度までの円弧をグラフィック表示します。計測 X 座標は外部参照データ番号 5、計測 Y 座標は外部参照データ番号 6、計測半径は外部参照データ番号 7 です。
位置ずれ修正結果にかかわらず固定位置に表示するため、<unitNo> パラメータには、当該処理を実行するユニットマクロ処理ユニット自身の処理ユニット番号を指定します。

```
*MEASUREDISPG

    Rem 計測結果を取得します
    GetUnitData 5, 5, X#
    GetUnitData 5, 6, Y#
    GetUnitData 5, 7, R#

    Rem 描画属性を設定します
    SetDrawStyle PS_SOLID,1,JUDGE_OK

    Rem グラフィックを描画します
    DrawArc Int(X#), Int(Y#), Int(R#), -90, 180, 0, UnitNo

Return
```

・使用可能なモジュール
ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「GetUnitData (P.8-138)」
「SetDrawStyle (P.8-250)」
「UnitNo (P.8-318)」
「Int (P.8-153)」
「UnitData (P.8-311)」
「Ut (P.8-320)」

DrawArcW

画像表示ウィンドウに幅付き円弧を描画します。

・書式
DrawArcW <x>, <y>, <radius>, <start>, <end>, <width>, <imageNo>[, <unitNo>]

・パラメータ

パラメータ名	データ型	説明
<x>	整数型	描画する幅付き円弧の中心 X 座標
<y>	整数型	描画する幅付き円弧の中心 Y 座標
<radius>	整数型	描画する幅付き円弧の半径
<start>	整数型	描画する幅付き円弧の開始点角度
<end>	整数型	描画する幅付き円弧の終了点角度
<width>	整数型	描画する幅付き円弧の幅
<imageNo>	整数型	描画対象の計測画像の番号（常に 0）
<unitNo>	整数型	描画を実行する処理ユニットの処理ユニット番号（0～（現在シーンの処理ユニット数 -1））

・戻り値
なし

・解説

<x> パラメータと<y> パラメータで指定した点を中心とする、<radius> パラメータで指定した半径の円の、<start> パラメータで指定した開始点角度から<end> パラメータで指定した終了点角度までの、<width> パラメータで指定した幅の幅付き円弧を、<imageNo> パラメータで指定した画像番号の計測画像に対してグラフィック描画します。
位置ずれ修正前の座標位置にグラフィックを描画したい場合は、<unitNo> パラメータに当該ユニットマクロの処理ユニット番号を指定してください。<unitNo> パラメータを省略した場合は、位置ずれ修正後の座標位置にグラフィックが描画されます。
<x> パラメータと<y> パラメータには、画像左上を原点とするカメラ座標を指定してください。
<start> パラメータと<end> パラメータには、カメラ座標 X 軸正方向を 0 度とする角度値を、右回りを正方向の角度として指定してください。
<imageNo> パラメータには、通常は 0 を指定してください。
<unitNo> パラメータには、通常は当該処理を実行するユニットマクロ処理ユニットの処理ユニット番号を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、*MEASUREDISPI サブルーチンまたは*MEASUREDISPG サブルーチンの中でしか使用できません。それ以外のサブルーチンで使用しようとした場合は、Illegal function call のエラーが発生します。
- 描画する前に、SetDrawStyle で線の種類や色や幅を設定してください。設定しなければ、直前の設定で描画されます。

・使用例

ユニットマクロ処理ユニットの*MEASUREDISPG サブルーチンの中で、処理ユニット番号 5 の円形スキャンエッジ位置処理ユニットで計測した位置に、計測した半径で、開始点角度-90 度から終了点角度 180 度までの円弧をグラフィック表示します。円弧は幅付き円弧で、計測した最大半径から最小半径の幅を持つ幅付き円弧を表示します。計測 X 座標は外部参照データ番号 5、計測 Y 座標は外部参照データ番号 6、計測半径は外部参照データ番号 7、最大半径は外部参照データ番号 8 番、最小半径は外部参照データ番号 9 番です。
位置ずれ修正結果にかかわらず固定位置に表示するため、<unitNo> パラメータには、当該処理を実行するユニットマクロ処理ユニット自身の処理ユニット番号を指定します。

```
*MEASUREDISPG

    Rem 計測結果を取得します
    GetUnitData 5, 5, X#
    GetUnitData 5, 6, Y#
    GetUnitData 5, 7, R#
    GetUnitData 5, 8, R_MAX#
    GetUnitData 5, 9, R_MIN#

    Rem 幅を求めます
    W# = R_MAX# - R_MIN#

    Rem 描画属性を設定します
    SetDrawStyle PS_SOLID,1,JUDGE_OK

    Rem グラフィックを描画します
    DrawArcW Int(X#), Int(Y#), Int(R#), -90, 180, Int(W#), 0, UnitNo

Return
```

・使用可能なモジュール
ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「GetUnitData (P.8-138)」
「SetDrawStyle (P.8-250)」
「UnitNo (P.8-318)」
「Int (P.8-153)」
「UnitData (P.8-311)」
「Ut (P.8-320)」

DrawBox

画像表示ウィンドウに長方形を描画します。

・書式
DrawBox <x0>, <y0>, <x1>, <y1>, <imageNo>[, <unitNo>]

・パラメータ

パラメータ名	データ型	説明
<x0>	整数型	描画する長方形の左上隅 X 座標
<y0>	整数型	描画する長方形の左上隅 Y 座標
<x1>	整数型	描画する長方形の右下隅 X 座標
<y1>	整数型	描画する長方形の右下隅 Y 座標
<imageNo>	整数型	描画対象の計測画像の番号（常に 0）
<unitNo>	整数型	描画を実行する処理ユニットの処理ユニット番号（0～（現在シーンの処理ユニット数 -1））

・戻り値
なし

・解説
<x0> パラメータと<y0> パラメータで指定した X 座標と Y 座標を左上隅の頂点として、<x1> パラメータと<y1> パラメータで指定した X 座標と Y 座標を右下隅の頂点とする長方形を、<imageNo> パラメータで指定した画像番号の計測画像に対してグラフィック描画します。

位置ずれ修正前の座標位置にグラフィックを描画したい場合は、<unitNo> パラメータに当該ユニットマクロの処理ユニット番号を指定してください。<unitNo> パラメータを省略した場合は、位置ずれ修正後の座標位置にグラフィックが描画されます。
<x0> パラメータと<y0> パラメータおよび<x1> パラメータと<y1> パラメータには、画像左上を原点とするカメラ座標を指定してください。
<imageNo> パラメータには、通常は 0 を指定してください。
<unitNo> パラメータには、通常は当該処理を実行するユニットマクロ処理ユニットの処理ユニット番号を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、*MEASUREDISPI サブルーチンまたは*MEASUREDISPG サブルーチンの中でしか使用できません。それ以外のサブルーチンで使用しようとした場合は、Illegal function call のエラーが発生します。
- 描画する前に、SetDrawStyle で線の種類や色や幅を設定してください。設定しなければ、直前の設定で描画されます。

・使用例

ユニットマクロ処理ユニットの*MEASUREDISPG サブルーチンの中で、処理ユニット番号 5 のサーチ処理ユニットで計測した位置と処理ユニット番号 6 のサーチ処理ユニットで計測した位置をそれぞれ左上隅座標、右下隅座標とする長方形をグラフィック表示します。計測 X 座標は外部参照データ番号 6、計測 Y 座標は外部参照データ番号 7 です。
位置ずれ修正結果にかかわらず固定位置に表示するため、<unitNo> パラメータには、当該処理を実行するユニットマクロ処理ユニット自身の処理ユニット番号を指定します。

```
*MEASUREDISPG

    Rem 計測結果を取得します
    GetUnitData 5, 6, X_LEFTTOP#
    GetUnitData 5, 7, Y_LEFTTOP#
    GetUnitData 6, 6, X_RIGHTBOTTOM#
    GetUnitData 6, 7, Y_RIGHTBOTTOM#

    Rem 描画属性を設定します
    SetDrawStyle PS_SOLID,1,JUDGE_OK

    Rem グラフィックを描画します
    DrawBox Int(X_LEFTTOP#), Int(Y_LEFTTOP#), Int(X_RIGHTBOTTOM#), Int(Y_RIGHTBOTTOM#), 0,
    UnitNo

Return
```

・使用可能なモジュール

ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「GetUnitData (P.8-138)」
「SetDrawStyle (P.8-250)」
「UnitNo (P.8-318)」
「Int (P.8-153)」
「UnitData (P.8-311)」
「Ut (P.8-320)」

DrawCircle

画像表示ウィンドウに円形を描画します。

・書式

DrawCircle <x>, <y>, <radius>, <imageNo>[, <unitNo>]

・パラメータ

パラメータ名	データ型	説明
<x>	整数型	描画する円の中心 X 座標
<y>	整数型	描画する円の中心 Y 座標
<radius>	整数型	描画する円の半径
<imageNo>	整数型	描画対象の計測画像の番号（常に 0）
<unitNo>	整数型	描画を実行する処理ユニットの処理ユニット番号（0～（現在シーンの処理ユニット数 -1)）

・戻り値

なし

・解説

<x> パラメータと<y> パラメータで指定した点を中心とする、<radius> パラメータで指定した半径の円を、<imageNo> パラメータで指定した画像番号の計測画像に対してグラフィック描画します。位置ずれ修正前の座標位置にグラフィックを描画したい場合は、<unitNo> パラメータに当該ユニットマクロの処理ユニット番号を指定してください。<unitNo> パラメータを省略した場合は、位置ずれ修正後の座標位置にグラフィックが描画されます。
<x> パラメータと<y> パラメータには、画像左上を原点とするカメラ座標を指定してください。
<imageNo> パラメータには、通常は 0 を指定してください。
<unitNo> パラメータには、通常は当該処理を実行するユニットマクロ処理ユニットの処理ユニット番号を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、*MEASUREDISPI サブルーチンまたは*MEASUREDISPG サブルーチンの中でしか使用できません。それ以外のサブルーチンで使用しようとした場合は、Illegal function call のエラーが発生します。
- 描画する前に、SetDrawStyle で線の種類や色や幅を設定してください。設定しなければ、直前の設定で描画されます。

・使用例

ユニットマクロ処理ユニットの*MEASUREDISPG サブルーチンの中で、処理ユニット番号 5 の円形スキャンエッジ位置処理ユニットで計測した位置に、計測した半径で円をグラフィック表示します。計測 X 座標は外部参照データ番号 5、計測 Y 座標は外部参照データ番号 6、計測半径は外部参照データ番号 7 です。

位置ずれ修正結果にかかわらず固定位置に表示するため、<unitNo> パラメータには、当該処理を実行するユニットマクロ処理ユニット自身の処理ユニット番号を指定します。

```
*MEASUREDISPG

    Rem 計測結果を取得します
    GetUnitData 5, 5, X#
    GetUnitData 5, 6, Y#
    GetUnitData 5, 7, R#

    Rem 描画属性を設定します
    SetDrawStyle PS_SOLID,1,JUDGE_OK

    Rem グラフィックを描画します
    DrawCircle Int(X#), Int(Y#), Int(R#), 0, UnitNo

Return
```

・使用可能なモジュール
ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「GetUnitData (P.8-138)」
「SetDrawStyle (P.8-250)」
「UnitNo (P.8-318)」
「Int (P.8-153)」
「UnitData (P.8-311)」
「Ut (P.8-320)」

DrawCircleW

画像表示ウィンドウに幅付き円形を描画します。

・書式
DrawCircleW <x>, <y>, <width>, <radius>, <imageNo>[, <unitNo>]

・パラメータ

パラメータ名	データ型	説明
<x>	整数型	描画する円の中心 X 座標
<y>	整数型	描画する円の中心 Y 座標
<width>	整数型	描画する幅付き円の幅
<radius>	整数型	描画する円の半径
<imageNo>	整数型	描画対象の計測画像の番号（常に 0）
<unitNo>	整数型	描画を実行する処理ユニットの処理ユニット番号（0～（現在シーンの処理ユニット数 -1））

・戻り値
なし

・解説
<x> パラメータと<y> パラメータで指定した点を中心とする、<width> パラメータで指定した幅を持つ、<radius> パラメータで指定した半径の円を、<imageNo> パラメータで指定した画像番号の計測画像に対してグラフィック描画します。

位置ずれ修正前の座標位置にグラフィックを描画したい場合は、<unitNo> パラメータに当該ユニットマクロの処理ユニット番号を指定してください。<unitNo> パラメータを省略した場合は、位置ずれ修正後の座標位置にグラフィックが描画されます。
<x> パラメータと<y> パラメータには、画像左上を原点とするカメラ座標を指定してください。
<imageNo> パラメータには、通常は 0 を指定してください。
<unitNo> パラメータには、通常は当該処理を実行するユニットマクロ処理ユニットの処理ユニット番号を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、*MEASUREDISPI サブルーチンまたは*MEASUREDISPG サブルーチンの中でしか使用できません。それ以外のサブルーチンで使用しようとした場合は、Illegal function call のエラーが発生します。
- 描画する前に、SetDrawStyle で線の種類や色や幅を設定してください。設定しなければ、直前の設定で描画されます。

・使用例

ユニットマクロ処理ユニットの*MEASUREDISPG サブルーチンの中で、処理ユニット番号 5 の円形スキャンエッジ位置処理ユニットで計測した位置に、計測した半径で円をグラフィック表示します。円は幅付き円形で、計測した最大半径から最小半径の幅を持つ幅付き円を表示します。計測 X 座標は外部参照データ番号 5、計測 Y 座標は外部参照データ番号 6、計測半径は外部参照データ番号 7、最大半径は外部参照データ番号 8 番、最小半径は外部参照データ番号 9 番です。
位置ずれ修正結果にかかわらず固定位置に表示するため、<unitNo> パラメータには、当該処理を実行するユニットマクロ処理ユニット自身の処理ユニット番号を指定します。

```
*MEASUREDISPG

    Rem 計測結果を取得します
    GetUnitData 5, 5, X#
    GetUnitData 5, 6, Y#
    GetUnitData 5, 7, R#
    GetUnitData 5, 8, R_MAX#
    GetUnitData 5, 9, R_MIN#

    Rem 幅を求めます
    W# = R_MAX# - R_MIN#

    Rem 描画属性を設定します
    SetDrawStyle PS_SOLID,1,JUDGE_OK

    Rem グラフィックを描画します
    DrawCircleW Int(X#), Int(Y#), Int(W#), Int(R#), 0, UnitNo

Return
```

・使用可能なモジュール

ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「GetUnitData (P.8-138)」
「SetDrawStyle (P.8-250)」
「UnitNo (P.8-318)」
「Int (P.8-153)」
「UnitData (P.8-311)」
「Ut (P.8-320)」

DrawCursor

画像表示ウィンドウに十字カーソルを描画します。

・書式

DrawCursor <x>, <y>, <imageNo>[, <unitNo>]

・パラメータ

パラメータ名	データ型	説明
<x>	整数型	描画する十字カーソルの中心 X 座標
<y>	整数型	描画する十字カーソルの中心 Y 座標
<imageNo>	整数型	描画対象の計測画像の番号（常に 0）
<unitNo>	整数型	描画を実行する処理ユニットの処理ユニット番号（0～（現在シーンの処理ユニット数 -1））

・戻り値

なし

・解説

<x> パラメータと<y> パラメータで指定した点を中心とする十字カーソルを、<imageNo> パラメータで指定した画像番号の計測画像に対してグラフィック描画します。
位置ずれ修正前の座標位置にグラフィックを描画したい場合は、<unitNo> パラメータに当該ユニットマクロの処理ユニット番号を指定してください。<unitNo> パラメータを省略した場合は、位置ずれ修正後の座標位置にグラフィックが描画されます。
<x> パラメータと<y> パラメータには、画像左上を原点とするカメラ座標を指定してください。
<imageNo> パラメータには、通常は 0 を指定してください。
<unitNo> パラメータには、通常は当該処理を実行するユニットマクロ処理ユニットの処理ユニット番号を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、*MEASUREDISPI サブルーチンまたは*MEASUREDISPG サブルーチンの中でしか使用できません。それ以外のサブルーチンで使用しようとした場合は、Illegal function call のエラーが発生します。
- 描画する前に、SetDrawStyle で線の種類や色や幅を設定してください。設定しなければ、直前の設定で描画されます。

・使用例

ユニットマクロ処理ユニットの*MEASUREDISPG サブルーチンの中で、処理ユニット番号 5 のサーチ処理ユニットで計測した位置に十字カーソルをグラフィック表示します。計測 X 座標は外部参照データ番号 6、計測 Y 座標は外部参照データ番号 7 です。

位置ずれ修正結果にかかわらず固定位置に表示するため、<unitNo> パラメータには、当該処理を実行するユニットマクロ処理ユニット自身の処理ユニット番号を指定します。

```
*MEASUREDISPG

    Rem 計測結果を取得します
    GetUnitData 5, 6, X#
    GetUnitData 5, 7, Y#

    Rem 描画属性を設定します
    SetDrawStyle PS_SOLID,1,JUDGE_OK

    Rem グラフィックを描画します
    DrawCursor Int(X#), Int(Y#), 0, UnitNo

Return
```

・使用可能なモジュール
ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「GetUnitData (P.8-138)」
「SetDrawStyle (P.8-250)」
「UnitNo (P.8-318)」
「Int (P.8-153)」
「UnitData (P.8-311)」
「Ut (P.8-320)」

DrawEllipse

画像表示ウィンドウに楕円形を描画します。

・書式
DrawEllipse <x>, <y>, <radiusX>, <radiusY>, <imageNo>[, <unitNo>]

・パラメータ

パラメータ名	データ型	説明
<x>	整数型	描画する楕円の中心 X 座標
<y>	整数型	描画する楕円の中心 Y 座標
<radiusX>	整数型	描画する楕円の X 方向の半径
<radiusY>	整数型	描画する楕円の Y 方向の半径
<imageNo>	整数型	描画対象の計測画像の番号（常に 0）
<unitNo>	整数型	描画を実行する処理ユニットの処理ユニット番号（0～（現在シーンの処理ユニット数 -1））

・戻り値
なし

・解説
<x> パラメータと<y> パラメータで指定した点を中心とする、<radiusX> パラメータで指定した X 軸方向の半径と<radiusY> パラメータで指定した Y 軸方向の半径の楕円を、<imageNo> パラメータで指定した画像番号の計測画像に対してグラフィック描画します。
位置ずれ修正前の座標位置にグラフィックを描画したい場合は、<unitNo> パラメータに当該ユニットマクロの処理ユニット番号を指定してください。<unitNo> パラメータを省略した場合は、位置ずれ修正後の座標位置にグラフィックが描画されます。

<x> パラメータと<y> パラメータには、画像左上を原点とするカメラ座標を指定してください。
<imageNo> パラメータには、通常は 0 を指定してください。
<unitNo> パラメータには、通常は当該処理を実行するユニットマクロ処理ユニットの処理ユニット番号を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、*MEASUREDISPI サブルーチンまたは*MEASUREDISPG サブルーチンの中でしか使用できません。それ以外のサブルーチンで使用しようとした場合は、Illegal function call のエラーが発生します。
- 描画する前に、SetDrawStyle で線の種類や色や幅を設定してください。設定しなければ、直前の設定で描画されます。

・使用例

ユニットマクロ処理ユニットの*MEASUREDISPG サブルーチンの中で、処理ユニット番号 5 のラベリング処理ユニットで計測した重心位置に、計測した楕円近似長径を X 方向の径、楕円近似短径を Y 方向の径とする楕円をグラフィック表示します。ここではラベリング処理ユニットの判定条件の設定を、0 番から順に重心 X、重心 Y、楕円近似長径、楕円近似短径として設定します。このとき重心 X 座標は外部参照データ番号 1000、重心 Y 座標は外部参照データ番号 1100、楕円近似長径は外部参照データ番号 1200、楕円近似短径は外部参照データ番号 1300 です。
位置ずれ修正結果にかかわらず固定位置に表示するため、<unitNo> パラメータには、当該処理を実行するユニットマクロ処理ユニット自身の処理ユニット番号を指定します。

```
*MEASUREDISPG

    Rem 計測結果を取得します
    GetUnitData 5, 1000, X#
    GetUnitData 5, 1100, Y#
    GetUnitData 5, 1200, XDIAMETER#
    GetUnitData 5, 1300, YDIAMETER#

    Rem 描画属性を設定します
    SetDrawStyle PS_SOLID,1,JUDGE_OK

    Rem グラフィックを描画します
    DrawEllipse Int(X#), Int(Y#), Int(XDIAMETER# / 2), Int(YDIAMETER# / 2), 0, UnitNo

Return
```

・使用可能なモジュール

ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「GetUnitData (P.8-138)」
「Int (P.8-153)」
「SetDrawStyle (P.8-250)」
「UnitData (P.8-311)」
「UnitNo (P.8-318)」
「Ut (P.8-320)」

DrawFigure

画像表示ウィンドウに図形を描画します。

・書式

DrawFigure <figure()>, <imageNo>[, <unitNo>]

・パラメータ

パラメータ名	データ型	説明
<figure()>	整数型配列	描画する図形の図形データを格納した 1 次元配列（参照：「9-6 図形データ一覧 (P.9-67)」）
<imageNo>	整数型	描画対象の計測画像の番号（常に 0）
<unitNo>	整数型	描画を実行する処理ユニットの処理ユニット番号（0～（現在シーンの処理ユニット数 -1））

・戻り値

なし

・解説

<figure()> パラメータで指定した図形を、<imageNo> パラメータで指定した画像番号の計測画像に対してグラフィック描画します。
位置ずれ修正前の座標位置にグラフィックを描画したい場合は、<unitNo> パラメータに当該ユニットマクロの処理ユニット番号を指定してください。<unitNo> パラメータを省略した場合は、位置ずれ修正後の座標位置にグラフィックが描画されます。
<figure()> パラメータには、図形データを格納する 1 次元の整数型配列変数を、要素番号を指定せず () だけ付加して指定してください。
<imageNo> パラメータには、通常は 0 を指定してください。
<unitNo> パラメータには、通常は当該処理を実行するユニットマクロ処理ユニットの処理ユニット番号を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、*MEASUREDISPI サブルーチンまたは*MEASUREDISPG サブルーチンの中でしか使用できません。それ以外のサブルーチンで使用しようとした場合は、Illegal function call のエラーが発生します。
- 描画する前に、SetDrawStyle で線の種類や色や幅を設定してください。設定しなければ、直前の設定で描画されます。

・使用例

ユニットマクロ処理ユニットの*MEASUREDISPG サブルーチンの中で、処理ユニット番号 5、6、7、8 のサーチ処理ユニットのモデル登録図形を一度に表示します。サーチ処理項目のモデル登録図形は、図形番号 0 です。（参照：「9-7 図形番号一覧 (P.9-70)」）
Dim 関数で定義する配列として、図形データの要素数以上の要素数を持つ配列を定義してください。

```
*MEASUREDISPG

    Dim FIGURE1&(255), FIGURE2&(255), FIGURE3&(255), FIGURE4&(255)

    Rem モデル図形の図形データを取得します
    GetUnitFigure 5, 0, FIGURE1&()
    GetUnitFigure 6, 0, FIGURE2&()
    GetUnitFigure 7, 0, FIGURE3&()
    GetUnitFigure 8, 0, FIGURE4&()

    Rem 描画属性を設定します
    SetDrawStyle PS_SOLID,1,JUDGE_OK

    Rem グラフィックを描画します
    DrawFigure FIGURE1&(), 0, UnitNo
    DrawFigure FIGURE2&(), 0, UnitNo
    DrawFigure FIGURE3&(), 0, UnitNo
    DrawFigure FIGURE4&(), 0, UnitNo

Return
```

・使用可能なモジュール
ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「GetUnitData (P.8-138)」
「Int (P.8-153)」
「SetDrawStyle (P.8-250)」
「UnitData (P.8-311)」
「UnitNo (P.8-318)」
「Ut (P.8-320)」

DrawFillImage

画像表示ウィンドウに塗り潰し画像を描画します。

・書式
DrawFillImage <color>

・パラメータ

パラメータ名	データ型	説明
<color>	整数型	塗り潰し色のカラー値（参照：「RGB (P.8-222)」）

・戻り値
なし

・解説
<color> パラメータで指定した色で塗り潰した画像を描画します。
<color> パラメータには、RGB 関数で取得したカラー値を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、*MEASUREDISPI サブルーチンまたは*MEASUREDISPG サブルーチンの中でしか使用できません。それ以外のサブルーチンで使用しようとした場合は、Illegal function call のエラーが発生します。

・使用例

ユニットマクロ処理ユニットの*MEASUREDISPI サブルーチンの中で、白色で塗り潰した画像を表示します。

```
*MEASUREDISPI

    DrawFillImage RGB(255, 255, 255)

Return
```

・使用可能なモジュール

ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「DrawMeasureImage (P.8-91)」　「DrawUnitImage (P.8-100)」
「RGB (P.8-222)」

DrawJudgeText

テキスト表示ウィンドウに判定結果の文字列を描画します。

・書式

DrawJudgeText <judge>

・パラメータ

パラメータ名	データ型	説明
<judge>	整数型	描画する判定結果 JUDGE_NC：「判定なし（未計測）」 JUDGE_OK：「判定：OK」 JUDGE_NG：「判定：NG」 JUDGE_IMAGEERROR：「判定：NG（画像不整合）」 JUDGE_MODELERROR：判定：NG（モデル未登録）」 JUDGE_MEMORYERROR：「判定：NG（メモリ不足）」 JUDGE_ERROR：「判定：NG（計測不能）」

・戻り値

なし

・解説

<judge> パラメータで指定した判定結果の文字列を、テキスト表示ウィンドウに描画します。
<judge> パラメータとして、UnitJudge 関数で取得した値を指定することもできます。（参照：「UnitJudge (P.8-317)」）
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定しても、Illegal function call のエラーは発生しません。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。

マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、*MEASUREDISPT サブルーチンの中でしか使用できません。それ以外のサブルーチンで使用しようとした場合は、Illegal function call のエラーが発生します。

・使用例

ユニットマクロ処理ユニットの*MEASUREDISPT サブルーチンの中で、処理ユニットの判定結果の文字列をテキスト表示ウィンドウに表示します。

```
*MEASUREDISPT

    DrawJudgeText UnitJudge(UnitNo)

Return
```

・使用可能なモジュール

ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「DrawText (P.8-97)」
「UnitData (P.8-311)」
「UnitNo (P.8-318)」
「GetUnitData (P.8-138)」
「UnitJudge (P.8-317)」

DrawLine

画像表示ウィンドウに直線を描画します。

・書式

DrawLine <x0>, <y0>, <x1>, <y1>, <imageNo>[, <unitNo>]

・パラメータ

パラメータ名	データ型	説明
<x0>	整数型	描画する直線の始点 X 座標
<y0>	整数型	描画する直線の始点 Y 座標
<x1>	整数型	描画する直線の終点 X 座標
<y1>	整数型	描画する直線の終点 Y 座標
<imageNo>	整数型	描画対象の計測画像の番号（常に 0）
<unitNo>	整数型	描画を実行する処理ユニットの処理ユニット番号（0～（現在シーンの処理ユニット数 -1））

・戻り値

なし

・解説

<x0> パラメータと<y0> パラメータで指定した X 座標と Y 座標を始点として、<x1> パラメータと<y1> パラメータで指定した X 座標と Y 座標を終点とする直線を、<imageNo> パラメータで指定した画像番号の計測画像に対してグラフィック描画します。

位置ずれ修正前の座標位置にグラフィックを描画したい場合は、<unitNo> パラメータに当該ユニットマクロの処理ユニット番号を指定してください。<unitNo> パラメータを省略した場合は、位置ずれ修正後の座標位置にグラフィックが描画されます。
<x0> パラメータと<y0> パラメータおよび<x1> パラメータと<y1> パラメータには、画像左上を原点とするカメラ座標を指定してください。
<imageNo> パラメータには、通常は 0 を指定してください。
<unitNo> パラメータには、通常は当該処理を実行するユニットマクロ処理ユニットの処理ユニット番号を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、*MEASUREDISPI サブルーチンまたは*MEASUREDISPG サブルーチンの中でしか使用できません。それ以外のサブルーチンで使用しようとした場合は、Illegal function call のエラーが発生します。
- 描画する前に、SetDrawStyle で線の種類や色や幅を設定してください。設定しなければ、直前の設定で描画されます。

・使用例

ユニットマクロ処理ユニットの*MEASUREDISPG サブルーチンの中で、処理ユニット番号 5 のサーチ処理ユニットで計測した位置と処理ユニット番号 6 のサーチ処理ユニットで計測した位置をそれぞれ始点、終点とする直線をグラフィック表示します。計測 X 座標は外部参照データ番号 6、計測 Y 座標は外部参照データ番号 7 です。
位置ずれ修正結果にかかわらず固定位置に表示するため、<unitNo> パラメータには、当該処理を実行するユニットマクロ処理ユニット自身の処理ユニット番号を指定します。

```
*MEASUREDISPG

    Rem 計測結果を取得します
    GetUnitData 5, 6, X_START#
    GetUnitData 5, 7, Y_START#
    GetUnitData 6, 6, X_END#
    GetUnitData 6, 7, Y_END#

    Rem 描画属性を設定します
    SetDrawStyle PS_SOLID,1,JUDGE_OK

    Rem グラフィックを描画します
    DrawLine Int(X_START#), Int(Y_START#), Int(X_END#), Int(Y_END#), 0, UnitNo

Return
```

・使用可能なモジュール

ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「GetUnitData (P.8-138)」　　「Int (P.8-153)」

「SetDrawStyle (P.8-250)」
「UnitNo (P.8-318)」
「UnitData (P.8-311)」
「Ut (P.8-320)」

DrawLineW

画像表示ウィンドウに幅付き直線を描画します。

・書式

DrawLineW <x0>, <y0>, <x1>, <y1>, <width>, <imageNo>[, <unitNo>]

・パラメータ

パラメータ名	データ型	説明
<x0>	整数型	描画する直線の始点 X 座標
<y0>	整数型	描画する直線の始点 Y 座標
<x1>	整数型	描画する直線の終点 X 座標
<y1>	整数型	描画する直線の終点 Y 座標
<width>	整数型	描画する直線の幅
<imageNo>	整数型	描画対象の計測画像の番号（常に 0）
<unitNo>	整数型	描画を実行する処理ユニットの処理ユニット番号（0～（現在シーンの処理ユニット数 -1））

・戻り値

なし

・解説

<x0> パラメータと<y0> パラメータで指定した X 座標と Y 座標を始点として、<x1> パラメータと<y1> パラメータで指定した X 座標と Y 座標を終点とする、<width> パラメータで指定した幅の直線を、<imageNo> パラメータで指定した画像番号の計測画像に対してグラフィック描画します。
位置ずれ修正前の座標位置にグラフィックを描画したい場合は、<unitNo> パラメータに当該ユニットマクロの処理ユニット番号を指定してください。<unitNo> パラメータを省略した場合は、位置ずれ修正後の座標位置にグラフィックが描画されます。
<x0> パラメータと<y0> パラメータおよび<x1> パラメータと<y1> パラメータには、画像左上を原点とするカメラ座標を指定してください。
<imageNo> パラメータには、通常は 0 を指定してください。
<unitNo> パラメータには、通常は当該処理を実行するユニットマクロ処理ユニットの処理ユニット番号を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、*MEASUREDISPI サブルーチンまたは*MEASUREDISPG サブルーチンの中でしか使用できません。それ以外のサブルーチンで使用しようとした場合は、Illegal function call のエラーが発生します。
- 描画する前に、SetDrawStyle で線の種類や色や幅を設定してください。設定しなければ、直前の設定で描画されます。

・使用例

ユニットマクロ処理ユニットの*MEASUREDISPG サブルーチンの中で、処理ユニット番号 5 のサーチ処理ユニットで計測した位置と処理ユニット番号 6 のサーチ処理ユニットで計測した位置をそれぞれ始点、終点とする幅 10 の幅付き直線をグラフィック表示します。計測 X 座標は外部参照データ番号 6、計測 Y 座標は外部参照データ番号 7 です。
位置ずれ修正結果にかかわらず固定位置に表示するため、<unitNo> パラメータには、当該処理を実行するユニットマクロ処理ユニット自身の処理ユニット番号を指定します。

```
*MEASUREDISPG

    Rem 計測結果を取得します
    GetUnitData 5, 6, X_START#
    GetUnitData 5, 7, Y_START#
    GetUnitData 6, 6, X_END#
    GetUnitData 6, 7, Y_END#

    Rem 描画属性を設定します
    SetDrawStyle PS_SOLID,1,JUDGE_OK

    Rem Rem グラフィックを描画します
    DrawLineW Int(X_START#), Int(Y_START#), Int(X_END#), Int(Y_END#), 10, 0, UnitNo

Return
```

・使用可能なモジュール
ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「GetUnitData (P.8-138)」
「Int (P.8-153)」
「SetDrawStyle (P.8-250)」
「UnitData (P.8-311)」
「UnitNo (P.8-318)」
「Ut (P.8-320)」

DrawMeasureImage

画像表示ウィンドウに計測画像を描画します。

・書式
DrawMeasureImage <imageNo>

・パラメータ

パラメータ名	データ型	説明
<imageNo>	整数型	描画対象の計測画像の番号（常に 0）

・戻り値
なし

・解説
<imageNo> パラメータで指定したこのマクロ関数を実行するユニットマクロ処理ユニットの計測画像番号の画像を描画します。
<imageNo> パラメータとして、存在しない数値を指定しても、エラーは発生しません。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。

整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、*MEASUREDISPI サブルーチンまたは*MEASUREDISPG サブルーチンの中でしか使用できません。それ以外のサブルーチンで使用しようとした場合は、Illegal function call のエラーが発生します。

・使用例

ユニットマクロ処理ユニットの*MEASUREDISPI サブルーチンの中で、計測画像を表示します。

```
*MEASUREDISPI

  DrawMeasureImage 0

Return
```

・使用可能なモジュール

ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「DrawFillImage (P.8-86)」　「DrawUnitImage (P.8-100)」

DrawPoint

画像表示ウィンドウに点を描画します。

・書式

DrawPoint <x>, <y>, <imageNo>[, <unitNo>]

・パラメータ

パラメータ名	データ型	説明
<x>	整数型	描画する点の X 座標
<y>	整数型	描画する点の Y 座標
<imageNo>	整数型	描画対象の計測画像の番号（常に 0）
<unitNo>	整数型	描画を実行する処理ユニットの処理ユニット番号（0～（現在シーンの処理ユニット数 -1））

・戻り値

なし

・解説

<x> パラメータと<y> パラメータで指定した位置に点を、<imageNo> パラメータで指定した画像番号の計測画像に対してグラフィック描画します。
位置ずれ修正前の座標位置にグラフィックを描画したい場合は、<unitNo> パラメータに当該ユニットマクロの処理ユニット番号を指定してください。<unitNo> パラメータを省略した場合は、位置ずれ修正後の座標位置にグラフィックが描画されます。
<x> パラメータと<y> パラメータには、画像左上を原点とするカメラ座標を指定してください。
<imageNo> パラメータには、通常は 0 を指定してください。

<unitNo> パラメータには、通常は当該処理を実行するユニットマクロ処理ユニットの処理ユニット番号を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、*MEASUREDISPI サブルーチンまたは*MEASUREDISPG サブルーチンの中でしか使用できません。それ以外のサブルーチンで使用しようとした場合は、Illegal function call のエラーが発生します。
- 描画する前に、SetDrawStyle で線の種類や色や幅を設定してください。設定しなければ、直前の設定で描画されます。

・使用例

ユニットマクロ処理ユニットの*MEASUREDISPG サブルーチンの中で、処理ユニット番号 5 のサーチ処理ユニットで計測した位置に点をグラフィック表示します。計測 X 座標は外部参照データ番号 6、計測 Y 座標は外部参照データ番号 7 です。
位置ずれ修正結果にかかわらず固定位置に表示するため、<unitNo> パラメータには、当該処理を実行するユニットマクロ処理ユニット自身の処理ユニット番号を指定します。

```
*MEASUREDISPG

    Rem 計測結果を取得します
    GetUnitData 5, 6, X#
    GetUnitData 5, 7, Y#

    Rem 描画属性を設定します
    SetDrawStyle PS_SOLID,1,JUDGE_OK

    Rem グラフィックを描画します
    DrawPoint Int(X#), Int(Y#), 0, UnitNo

Return
```

・使用可能なモジュール

ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「GetUnitData (P.8-138)」
「Int (P.8-153)」
「SetDrawStyle (P.8-250)」
「UnitData (P.8-311)」
「UnitNo (P.8-318)」
「Ut (P.8-320)」

DrawPolygon

画像表示ウィンドウに多角形を描画します。

・書式

DrawPolygon <count>, <x()>, <y()>, <imageNo>[, <unitNo>]

・パラメータ

パラメータ名	データ型	説明
<count>	整数型	描画する多角形の頂点の個数（0 ～）
<x()>	整数型配列	描画する多角形の X 座標を格納した 1 次元配列
<y()>	整数型配列	描画する多角形の Y 座標を格納した 1 次元配列
<imageNo>	整数型	描画対象の計測画像の番号（常に 0）
<unitNo>	整数型	描画を実行する処理ユニットの処理ユニット番号（0～（現在シーンの処理ユニット数 -1））

・戻り値

なし

・解説

<count> パラメータで指定した頂点の個数を持つ、<x()> パラメータと<y()> パラメータで指定した複数の X 座標と Y 座標を頂点とする多角形を、<imageNo> パラメータで指定した画像番号の計測画像に対してグラフィック描画します。
位置ずれ修正前の座標位置にグラフィックを描画したい場合は、<unitNo> パラメータに当該ユニットマクロの処理ユニット番号を指定してください。<unitNo> パラメータを省略した場合は、位置ずれ修正後の座標位置にグラフィックが描画されます。
<x()> パラメータと<y()> パラメータには、<count> パラメータで指定した以上の個数の座標値を格納した 1 次元の整数型配列変数を、要素番号を指定せず() だけ付加して指定してください。
<x> パラメータと<y> パラメータには、画像左上を原点とするカメラ座標を指定してください。
<imageNo> パラメータには、通常は 0 を指定してください。
<unitNo> パラメータには、通常は当該処理を実行するユニットマクロ処理ユニットの処理ユニット番号を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、*MEASUREDISPI サブルーチンまたは*MEASUREDISPG サブルーチンの中でしか使用できません。それ以外のサブルーチンで使用しようとした場合は、Illegal function call のエラーが発生します。
- 描画する前に、SetDrawStyle で線の種類や色や幅を設定してください。設定しなければ、直前の設定で描画されます。

・使用例

ユニットマクロ処理ユニットの*MEASUREDISPG サブルーチンの中で、処理ユニット番号 5、6、7 のサーチ処理ユニットで計測した位置を結ぶ多角形をグラフィック表示します。計測 X 座標は外部参照データ番号 6、計測 Y 座標は外部参照データ番号 7 です。
位置ずれ修正結果にかかわらず固定位置に表示するため、<unitNo> パラメータには、当該処理を実行するユニットマクロ処理ユニット自身の処理ユニット番号を指定します。

```
*MEASUREDISPG

    Dim X&(2), Y&(2)

    Rem 計測結果を取得します
    GetUnitData 5, 6, X&(0)
    GetUnitData 5, 7, Y&(0)
    GetUnitData 6, 6, X&(1)
    GetUnitData 6, 7, Y&(1)
    GetUnitData 7, 6, X&(2)
    GetUnitData 7, 7, Y&(2)

    Rem 描画属性を設定します
    SetDrawStyle PS_SOLID,1,JUDGE_OK

    Rem グラフィックを描画します
    DrawPolygon 3, X&(), Y&(), 0, UnitNo

Return
```

・使用可能なモジュール
ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「GetUnitData (P.8-138)」
「SetDrawStyle (P.8-250)」
「UnitNo (P.8-318)」
「Int (P.8-153)」
「UnitData (P.8-311)」
「Ut (P.8-320)」

DrawSearchFigure

画像表示ウィンドウにサーチ図形を描画します。

・書式
DrawSearchFigure <figure()>, <referenceX>, <referenceY>, <measureX>, <measureY>, <measureAngle>, <imageNo>, <unitNo>

・パラメータ

パラメータ名	データ型	説明
<figure()>	整数型配列	描画する図形の図形データを格納した 1 次元配列（参照：「9-6 図形データ一覧 (P.9-67)」）
<referenceX>	実数型	描画する図形の位置を特定するための検出点 X 座標
<referenceY>	実数型	描画する図形の位置を特定するための検出点 Y 座標
<measureX>	実数型	描画する図形の位置を特定するための計測 X 座標
<measureY>	実数型	描画する図形の位置を特定するための計測 Y 座標
<measureAngle>	実数型	描画する図形の角度を特定するための計測角度
<imageNo>	整数型	描画対象の計測画像の番号（常に 0）
<unitNo>	整数型	描画を実行する処理ユニットの処理ユニット番号（0～（現在シーンの処理ユニット数 -1））

・戻り値
なし

・解説

<figure()> パラメータで指定した図形を、<referenceX> パラメータと<referenceY> パラメータおよび<measureX> パラメータと<measureY> パラメータで特定した位置に、<measureAngle> パラメータで指定した角度で、<imageNo> パラメータで指定した画像番号の計測画像に対してグラフィック描画します。

このマクロ関数は、おもにサーチ処理ユニットや形状サーチ III 処理ユニットなどの、モデル登録図形と検出点を持つ処理ユニットの図形をグラフィック描画することができます。モデル登録図形または検出点を設定する機能を持たない処理ユニットの図形を描画する場合は、正しくグラフィック描画できません。

<referenceX> パラメータと<referenceY> パラメータには、通常はサーチ処理ユニットの設定パラメータである検出点 X と検出点 Y を指定してください。

<measureX> パラメータ、<measureY> パラメータおよび<measureAngle> パラメータには、通常はサーチ処理ユニットの計測座標 X、計測座標 Y および計測角度を指定してください。

位置ずれ修正前の座標位置にグラフィックを描画したい場合は、<unitNo> パラメータに当該ユニットマクロの処理ユニット番号を指定してください。<unitNo> パラメータを省略した場合は、位置ずれ修正後の座標位置にグラフィックが描画されます。

<figure()> パラメータには、図形データを格納した 1 次元の整数型配列変数を、要素番号を指定せず () だけ付加して指定してください。

<imageNo> パラメータには、通常は 0 を指定してください。

<unitNo> パラメータには、通常は当該処理を実行するユニットマクロ処理ユニットの処理ユニット番号を指定してください。

このマクロ関数は、おもにサーチ処理ユニットで検出したモデル図形を描画するときに使用します。

パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。

パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。

整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。

実数型のパラメータとして、-1.0E30～1.0E30 の範囲を超える値を指定した場合は、Overflow のエラーが発生する可能性があります。

マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、*MEASUREDISPI サブルーチンまたは*MEASUREDISPG サブルーチンの中でしか使用できません。それ以外のサブルーチンで使用しようとした場合は、Illegal function call のエラーが発生します。
- 描画する前に、SetDrawStyle で線の種類や色や幅を設定してください。設定しなければ、直前の設定で描画されます。

・使用例

ユニットマクロ処理ユニットの*MEASUREDISPG サブルーチンの中で、処理ユニット番号 5 のサーチ処理ユニットのモデル図形を表示します。計測 X 座標は外部参照データ番号 6、計測 Y 座標は外部参照データ番号 7、計測角度は外部参照データ番号 8、設定データの検出点 X 座標は外部参照データ番号 132、設定データの検出点 Y 座標は外部参照データ番号 133 です。

位置ずれ修正結果にかかわらず固定位置に表示するため、<unitNo> パラメータには、当該処理を実行するユニットマクロ処理ユニット自身の処理ユニット番号を指定します。

```
*MEASUREDISPG

    Dim FIGURE&(255)

    Rem モデル図形を取得します
    GetUnitFigure 5, 0, FIGURE&()

    Rem 計測結果を取得します
    GetUnitData 5, 6, X#
    GetUnitData 5, 7, Y#
    GetUnitData 5, 8, TH#
    GetUnitData 5, 132, RX#
    GetUnitData 5, 133, RY#

    Rem 描画属性を設定します
    SetDrawStyle PS_SOLID,1,JUDGE_OK

    Rem グラフィックを描画します
    DrawSearchFigure FIGURE&(), RX#, RY#, X#, Y#, TH#, 0, UnitNo

Return
```

・使用可能なモジュール
ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「GetUnitData (P.8-138)」
「Int (P.8-153)」
「SetDrawStyle (P.8-250)」
「UnitData (P.8-311)」
「UnitNo (P.8-318)」
「Ut (P.8-320)」

DrawText

テキスト表示ウィンドウに文字列を描画します。

・書式
DrawText <string>, <color>, <newLine>

・パラメータ

パラメータ名	データ型	説明
<string>	文字列型	描画する文字列
<color>	整数型	描画する文字列の色のカラー値 JUDGE_NC：未計測色（灰色） JUDGE_OK：OK 色（緑色） JUDGE_NG：NG 色（赤色） RGB 関数：任意の色
<newLine>	整数型	描画後の改行 0：改行しない 1：改行する

・戻り値
なし

・解説

<string> パラメータで指定した文字列を、<color> パラメータで指定した色を使って、<newLine> パラメータで指定した改行方法でテキスト表示ウィンドウに描画します。
<color> パラメータとして、RGB 関数で取得したカラー値を設定することもできます。(参照:「RGB (P.8-222)」)
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして、存在しない数値を指定しても、エラーは発生しません。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、*MEASUREDISPT サブルーチンの中でしか使用できません。それ以外のサブルーチンで使用しようとした場合は、Illegal function call のエラーが発生します。
- プログラム入力エリアでは、注釈コメントが入力できます。
  設定できる文字列は、以下のとおりです。
  「Language setting」で設定した言語と英語が設定可能です。
  詳細は、「注釈コメント (P.4-4)」を参照してください。

・使用例

ユニットマクロ処理ユニットの*MEASUREDISPT サブルーチンの中で、処理ユニット番号 5 のサーチ処理ユニットの計測結果をテキスト表示ウィンドウに表示します。計測 X 座標は外部参照データ番号 6、計測 Y 座標は外部参照データ番号 7、計測角度は外部参照データ番号 8 です。

```
*MEASUREDISPT

    Rem 計測結果を取得します
    GetUnitData 5, 6, X#
    GetUnitData 5, 7, Y#
    GetUnitData 5, 8, TH#

    Rem 文字列を OK 色改行なしで描画したあと区切り文字を描画します
    DrawText Str2$(X#, 4, 4, 0, 0), JUDGE_OK, 0
    DrawText ", ", JUDGE_OK, 0

    Rem 文字列を OK 色改行ありで描画します
    DrawText Str2$(Y#, 4, 4, 0, 0), JUDGE_OK, 1

    Rem 文字列を OK 色改行なしで描画します
    DrawText Str2$(TH#, 4, 4, 0, 0), JUDGE_OK, 0

Return
```

結果は以下のとおりになります。

```
123.4567, 10.5000
90.0000
```

・使用可能なモジュール

ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「DrawJudgeText (P.8-87)」
「RGB (P.8-222)」
「Str2$ (P.8-288)」
「GetUnitData (P.8-138)」
「Str$ (P.8-287)」
「UnitData (P.8-311)」

DrawTextG

画像表示ウィンドウに文字列を描画します。

・書式

DrawTextG <string>, <x>, <y>, <imageNo>[, <unitNo>]

・パラメータ

パラメータ名	データ型	説明
<string>	文字列型	描画する文字列
<x>	整数型	描画位置の先頭左上 X 座標
<y>	整数型	描画位置の先頭左上 Y 座標
<imageNo>	整数型	描画対象の計測画像の番号（常に 0）
<unitNo>	整数型	描画を実行する処理ユニットの処理ユニット番号（0～（現在シーンの処理ユニット数 -1)）

・戻り値

なし

・解説

<string> パラメータで指定した文字列を、<x> パラメータと<y> パラメータで指定した位置に、<imageNo> パラメータで指定した画像番号の計測画像に対してグラフィック描画します。
画像表示ウィンドウに% を表示したい場合は<string> パラメータに%% と指定します。%単体では使用しないでください。(使い方によって文字化けやエラーが発生します)
位置ずれ修正前の座標位置にグラフィックを描画したい場合は、<unitNo> パラメータに当該ユニットマクロの処理ユニット番号を指定してください。<unitNo> パラメータを省略した場合は、位置ずれ修正後の座標位置にグラフィックが描画されます。
<x> パラメータと<y> パラメータには、画像左上を原点とするカメラ座標を指定してください。
<imageNo> パラメータには、通常は 0 を指定してください。
<unitNo> パラメータには、通常は当該処理を実行するユニットマクロ処理ユニットの処理ユニット番号を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、*MEASUREDISPI サブルーチンまたは*MEASUREDISPG サブルーチンの中でしか使用できません。それ以外のサブルーチンで使用しようとした場合は、Illegal function call のエラーが発生します。
- プログラム入力エリアでは、注釈コメントが入力できます。
  設定できる文字列は、以下のとおりです。
  「Language setting」で設定した言語と英語が設定可能です。

詳細は、「注釈コメント (P.4-4)」を参照してください。

・使用例

ユニットマクロ処理ユニットの*MEASUREDISPG サブルーチンの中で、文字列"OK" を座標(100, 100) の固定位置にグラフィック表示します。位置ずれ修正結果にかかわらず固定位置に表示するため、<unitNo> パラメータには、当該処理を実行するユニットマクロ処理ユニット自身の処理ユニット番号を指定します。

```
*MEASUREDISPG

   DrawTextG "OK", 100, 100, 0, UnitNo

Return
```

・使用可能なモジュール

ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「GetUnitData (P.8-138)」
「Str$ (P.8-287)」
「UnitData (P.8-311)」
「Ut (P.8-320)」
「SetTextStyle (P.8-272)」
「Str2$ (P.8-288)」
「UnitNo (P.8-318)」

DrawUnitImage

画像表示ウィンドウに処理ユニット画像を描画します。

・書式

DrawUnitImage <unitNo>, <imageNo>

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	描画を実行する処理ユニットの処理ユニット番号 (0～ (現在シーンの処理ユニット数 -1))
<imageNo>	整数型	描画対象の計測画像の番号（参照：「9-9 画像番号一覧 (P.9-74)」)

・戻り値

なし

・解説

<unitNo> パラメータで指定した処理ユニットが持つ、<imageNo> パラメータで指定した画像番号の画像を描画します。
<unitNo> パラメータとして、存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
<imageNo> パラメータとして、存在しない数値を指定しても、エラーは発生しません。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、*MEASUREDISPI サブルーチンまたは*MEASUREDISPG サブルーチンの中でしか使用できません。それ以外のサブルーチンで使用しようとした場合は、Illegal function call のエラーが発生します。

・使用例

ユニットマクロ処理ユニットの*MEASUREDISPG サブルーチンの中で、あらかじめ出力画像 1 を設定しておいた処理ユニット番号 5 の高機能前処理ユニットの画像番号 1 の画像を表示します。

```
*MEASUREDISPG

    Rem 処理ユニット画像を表示します
    DrawUnitImage 5, 1

Return
```

・使用可能なモジュール

ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「DrawFillImage (P.8-86)」
「DrawMeasureImage (P.8-91)」
「UnitNo (P.8-318)」
「Ut (P.8-320)」

Dskf

ドライブの空き容量を求めます。

・書式

Dskf(<driveName>)

・パラメータ

パラメータ名	データ型	説明
<driveName>	文字列型	空き容量を求めるドライブ名

・戻り値

指定したドライブの空き容量を実数型の値で返します。
指定したドライブが存在しない場合は、-1 を返します。

・解説

<driveName> パラメータで指定したドライブの空き容量のバイト数を求めます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

ドライブの空き容量が 1KB（1024 バイト）未満になったら ERROR 信号を ON します。

```
Rem ドライブの空き容量をチェックします
DiskSize# = Dskf("E:¥")
If DiskSize# < 1024 Then
   Rem ERROR 信号を ON します
   PutPort "Parallello", 103, 1
Endif
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「Fcopy (P.8-115)」
「Kill (P.8-160)」
「PutPort (P.8-205)」
「Isfile (P.8-154)」
「Mkdir (P.8-188)」
「Rmdir (P.8-224)」

ElapsedTime

計測開始からの経過時間を取得します。

・書式

ElapsedTime(<mode>)

・パラメータ

パラメータ名	データ型	説明
<mode>	整数型	取得する経過時間の単位 0：ms 単位 1：μs 単位

・戻り値

整数型の経過時間の値を返します。

・解説

<mode> パラメータで指定した単位で、計測開始からの経過時間を取得します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

計測開始からユニットマクロ処理ユニットの MEASUREPROC サブルーチンでこのマクロ関数を実行するまでの経過時間を取得して、経過時間が 1000ms 以上の場合はテキスト表示ウィンドウにエラー文字列を表示します。

```
*MEASUREPROC

    Rem 経過時間を取得します
    TIME& = ElapsedTime(0)

    Return


*MEASUREDISPT

    Rem 経過時間が 1000ms 以上の場合は、エラー文字列を NG 色で表示します
    If TIME& > 999 Then
        DrawText "Error", JUDGE_NG, 1
    Endif

Return
```

・使用可能なモジュール

ユニット演算マクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「DrawText (P.8-97)」
「StartTimer (P.8-285)」
「Timer (P.8-298)」
「Wait (P.8-328)」

Eof

ファイルの終わりを調べます。

・書式

Eof(<fileNo>)

・パラメータ

パラメータ名	データ型	説明
<fileNo>	整数型	終わりを調べるファイルのファイル番号（0～15）

・戻り値

整数型のファイルの終端に達したどうかの値を返します。

- -1：ファイルの終端に達した
- 0：ファイルの終端に達していない

・解説

<fileNo> パラメータで指定したファイル番号のファイルの終端に達したかどうかを調べます。
<fileNo> パラメータとして、開かれていないファイル番号を指定した場合は、Illegal function call のエラーが発生します。
<fileNo> パラメータとして、0～15 の範囲を超える値を指定した場合は、Illegal function call のエラーが発生します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

ファイルの終端に達するまで改行コードで区切られたデータを読み込みます。

```
Dim ALLDATA$(255)

Rem ファイルを開きます
Open "E:¥input.dat" For Input As #1

For I&=0 to 255
   Rem ファイル先頭から改行コードで区切られた行を 1 行ずつ読み込みます
   Input #1, DATA$
   ALLDATA$(I&) = DATA$

   Rem ファイルの終端に達したかどうかチェックします
   If Eof(1) <> 0 Then
      Exit For
   Endif
Next

Rem ファイルを閉じます
Close #1
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「Close (P.8-45)」
「Input# (P.8-149)」
「Open For Append As# (P.8-192)」
「Open For Input As# (P.8-193)」
「Open For Output As# (P.8-194)」

Erase

配列変数を解放します。

・書式
Erase <array>[, <array>...]

・パラメータ

パラメータ名	データ型	説明
<array>	-	解放する配列変数

・戻り値
なし

・解説
<array> パラメータで指定した Dim 関数であらかじめ定義した配列変数のメモリ領域を解放します。このマクロ関数を実行して一時的に使用する配列変数を解放することで、メモリ領域を解放してメモリを有効活用することができます。
配列変数を解放せずに再定義した場合は、以前に定義した配列変数を解放してから再定義します。
パラメータとして、配列変数以外の変数を指定した場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
定義した配列を解放します。

```
Dim XY&(3)
Dim XY#(7, 15)
Dim CHARA$(31, 63, 127, 255)

Rem 配列変数を解放します
Erase XY&(), XY#(), CHARA$()
```

・使用可能なモジュール
ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「Dim (P.8-67)」

Errcmnd$

エラーが発生したマクロ関数名を取得します。

・書式
Errcmnd$

・パラメータ
なし

・戻り値
文字列型のエラーが発生したマクロ関数名の大文字アルファベット文字列の値を返します。

・解説
プログラムでエラーが発生したときの、エラーが発生したマクロ関数の名前の文字列を取得します。
エラーが発生していない場合やゼロ割などマクロ関数の処理以外でエラーが発生した場合は、ヌル文字列("") を返します。
このマクロ関数は、Try Catch End Try 関数の中で使用してください。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
ユニットマクロの*MEASUREPROC サブルーチンの中で、GetUnitData 関数でエラーが発生した場合だけ、システム状態監視コンソールウィンドウにエラー情報を出力して、それ以外のマクロ関数でエラーが発生した場合は何もせず計測処理を終了します。

```
*MEASUREPROC

   Try
      GetUnitData 5, 5, CR#

      SetUnitData 6, 143, CR#
   Catch
      If Errcmnd$ = "GetUnitData" Then
         Print Errcmnd$
      Endif
   End Try

Return
```

・使用可能なモジュール
ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「Errno (P.8-107)」
「Print (P.8-202)」
「Try Catch End Try (P.8-307)」
「GetUnitData (P.8-138)」
「SetUnitData (P.8-275)」

Errno

エラー番号を取得します。

・書式
Errno

・パラメータ
なし

・戻り値
整数型のエラー番号の値を返します。(参照：「9-1 エラー一覧 (P.9-2)」)

・解説
プログラムでエラーが発生したときの、エラー番号を取得します。
このマクロ関数は、Try Catch End Try 関数の中で使用してください。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
ユニットマクロの*MEASUREPROC サブルーチンの中で、Try Catch End Try 関数でエラーの発生を検知してエラー番号を取得します。

```
*MEASUREPROC

   Try
      WORK& = 0
      SUMM& = 100 + 200 + 300
      ANS& = SUMM& / WORK&
   Catch
      If Errno = 11 Then
         Rem エラー番号とエラーの内容をシステム状態監視コンソールウィンドウに出力します
         Print "Error Number = " + Str$(Errno) + ", Division by Zero"
      Endif
   End Try

Return
```

・使用可能なモジュール
ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「Errcmnd$ (P.8-106)」
「Try Catch End Try (P.8-307)」
「Print (P.8-202)」

ErrorOut

ERROR 信号の出力状態を設定します。

・書式

ErrorOut [<ioIdent>,] <errorKind>

・パラメータ

パラメータ名	データ型	説明
<ioIdent>	文字列型	使用する通信モジュールの識別名 （参照：「9-4 IO モジュール一覧 (P.9-25)」）
<errorKind>	整数型	エラー種別番号を設定します。0 以上の値を設定すると ERROR 信号が ON します。 -1：ERROR 信号 OFF 0：システム異常 1：システム異常（ファン／電圧異常） 3：バッテリ異常 4：UWF オーバレイ枯渇 10：カメラ接続異常 11：接続カメラに変更あり 12：カメラ過電流検出 13：照明機器接続構成異常 20：ロギングディスク書込異常 30：パラレル出力タイムアウト 31：PLC リンク通信エラー 32：パラレル I/O 過電流検出 40：データロード異常 41：データ転送異常 42：起動時シーングループ番号不正 43：起動時シーン番号不正

・戻り値

なし

・解説

<ioIdent> パラメータで指定した通信モジュールの ERROR 信号の状態を設定します。通信モジュールが EtherCAT の場合は、ERROR 信号の状態とエラー種別番号を設定します。

<ioIdent> パラメータとして、" "（空の文字列を指定する）、または、<ioIdent> パラメータを省略した場合は、すべての通信モジュールが対象となります。

<errorKind> パラメータが 0 以上の場合 ERROR 信号を ON、-1 の場合 ERROR 信号を OFF します。

パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。

パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定しても、Illegal function call のエラーは発生しません。

整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。

文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。

マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- *ErrorProc を記述すると、エラー発生時に記述された内容のみの動作になります。
  通常のエラー処理（使用例を参照）
  RaiseErrorProcEvent ERRORKIND&, 0, True

ErrorOut "", ERRORKIND&

をあわせて記述しておくことで、特定のエラー以外が発生した場合に、通常通りのエラー処理が行われるようになります。

・使用例

```
Rem ロギングエラーでメッセージボックスを表示
*ERRORPROC

If ERRORKIND& = 20 Then
   MessageBox "error"

Rem ロギングエラー以外のエラーの場合、通常のエラー処理をおこなう
Else
   RaiseErrorProcEvent ERRORKIND&, 0, True
   ErrorOut "", ERRORKIND&
EndIf

Return
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 5.40 以降

・関連項目

「MessageBox (P.8-185)」　「RaiseErrorProcEvent (P.8-207)」

ExecuteErrorProc

指定した種別のエラーが発生したことをシステムに通知します。

・書式

ExecuteErrorProc <errorKind>, <parameter>

・パラメータ

パラメータ名	データ型	説明
<errorKind>	整数型	通知するエラー処理イベントのエラー種別番号 0：システム異常 1：システム異常（ファン／電圧異常） 3：バッテリ異常 4：UWF オーバレイ枯渇 10：カメラ接続異常 11：接続カメラに変更あり 12：カメラ過電流検出 13：照明機器接続構成異常 20：ロギングディスク書込異常 30：パラレル出力タイムアウト 31：PLC リンク通信エラー 32：パラレル I/O 過電流検出 40：データロード異常 41：データ転送異常 42：起動時シーングループ番号不正 43：起動時シーン番号不正
<parameter>	整数型	通知するエラー処理イベントのパラメータ 現時点では未使用。そのため、常に 0 を指定する。

・戻り値

なし

・解説

指定した種別のエラーが発生したことをシステムに通知します。
独自のエラー種別番号を定義して、そのエラー種別番号に対するエラー処理をユニットマクロの *ErrorProc サブルーチン等に定義することで、独自のエラー処理を作成できます。
独自のエラー種別番号は、10000 以降の番号を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定しても、Illegal function call のエラーは発生しません。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- *MCRINIT に記述しないでください。

・使用例

OCR 取得文字が 0 のとき、エラーが発生したことをシステムに通知し、独自のエラー種別番号を定義します。そのエラー種別番号に対するエラー処理（エラーメッセージ）を作成します。

```
Rem ユニット 1 にある OCR 取得文字が 0 の時エラーを発生させます
■ユニットマクロ
GetUnitData 1, "characterNum0", NUM0&
SetGlobalData "NUM", NUM0&

■通信コマンドマクロ 1 (CMD0000)
Measure 1

GetGlobalData "NUM", NUM0&

If NUM0& = 0 Then
   ExecuteErrorProc 10000, 0
EndIf

■通信コマンドマクロ 2 (CMD0001)
*ERRORPROC

Rem 独自にエラー定義(10000)
If ERRORKIND& = 10000 Then
   MessageBox "ERROR"
   ErrorOut "", 10000
Else
   Rem エラー番号 10000 以外のエラーが発生した場合、通常のエラー処理を行う
   RaiseErrorProcEvent ERRORKIND&, 0, True
   ErrorOut "", ERRORKIND&
EndIf

Return
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン

バージョン 5.40 以降

・関連項目

「ErrorOut (P.8-108)」
「GetGlobalData (P.8-121)」
「GetUnitData (P.8-138)」
「Measure (P.8-176)」
「RaiseErrorProcEvent (P.8-207)」
「SetGlobalData (P.8-253)」
「SetUnitData (P.8-275)」

ExecuteImageLogging

画像ロギングを実行します。

・書式

ExecuteImageLogging <directory>, <header>

・パラメータ

パラメータ名	データ型	説明
<directory>	文字列型	ロギング画像を保存するサブフォルダ名
<header>	文字列型	ロギング画像のファイル名に付加するヘッダ文字列

・戻り値

なし

・解説

画像ロギングを実行して、<directory> パラメータで指定したサブフォルダに、<header> パラメータで指定したヘッダ文字列を計測 ID の文字列に付加したファイル名でロギング画像を保存します。
ロギング画像はシステム設定で指定したファイル形式（IFZ、BMP(BFZ)、JPG(JFZ)、PNG(PFZ)）で保存することができます。
ロギング画像は、システム設定のロギング設定で設定した画像ロギングの保存先フォルダに保存されます。（詳細は、『画像処理システム FH/FHV シリーズ ユーザーズマニュアル(SDNB-719)』の「ロギング条件を設定する［ロギング設定］」を参照してください。）
<directory>　パラメータでサブフォルダ名を指定した場合は、画像ロギングの保存先フォルダの中にサブフォルダが作成され、サブフォルダの中にロギング画像のファイルが保存されます。
<directory> パラメータで空の文字列" " を指定した場合は、サブフォルダは作成されません。
<header> パラメータでヘッダ文字列を指定した場合は、ファイル名の先頭にヘッダ文字列が付加されたファイル名で保存されます。<header> パラメータで空の文字列" " を指定した場合は、ロギング画像のファイル名にヘッダ文字列は付加されません。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
<directory> パラメータとして、63 文字を超える文字列を指定した場合は、64 文字め以降を切り捨てた 63 文字の文字列として扱われます。
<header> パラメータとして、31 文字を超える文字列を指定した場合は、32 文字め以降を切り捨てた 31 文字の文字列として扱われます。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、*MEASUREPROC サブルーチンの中でしか使用できません。それ以外のサブルーチンで使用しようとした場合は、マクロ関数の処理は実行されずエラーも発生しません。

・使用例

ユニットマクロ処理ユニットの MEASUREPROC サブルーチンで、システム設定でロギング先に設定したフォルダ"E:¥" 下のサブフォルダ"Image" に、ヘッダ"new_" が付加されたロギング画像を保存します。

```
*MEASUREPROC

    Rem 画像ロギングを実行します
    ExecuteImageLogging "Image", "new_"

Return
```

計測実行後に、"E:¥image" フォルダに"new_2012-11-01_13-11-25-0025.ifz" のファイルが保存されます。

・使用可能なモジュール
ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「GetSystemData (P.8-133)」
「GetUnitData (P.8-138)」
「Str$ (P.8-287)」
「Str2$ (P.8-288)」

ExitFzProcess

センサコントローラを終了させます。

・書式
ExitFzProcess

・パラメータ
なし

・戻り値
なし

・解説
FH のプロセスを終了します。
センサコントローラ上でこのマクロ関数を実行した場合は、実行後にセンサコントローラの電源を OFF してください。

・使用上の注意
なし

・使用例
本体へデータを保存後、センサコントローラを終了させます。

```
Rem 本体に保存を実行します
SaveData

Rem センサコントローラを終了させます
ExitFzProcess
```

・使用可能なモジュール
シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「SaveData (P.8-227)」
「SystemReset (P.8-294)」

Exp

自然対数の底 e を底とする指数関数の値を求めます。

・書式

Exp(<expression>)

・パラメータ

パラメータ名	データ型	説明
<expression>	整数型 実数型	指数を求める数式

・戻り値

実数型の指数（べき乗）計算した値を返します。

・解説

<expression>パラメータで指定した数式で、自然対数の底 e を指数（べき乗）計算した値を求めます。
Exp 関数の逆関数は Log 関数で求めることができます。Exp 関数を使用して、双曲正弦関数など他の数学上の関数を導き出すことができます。
<expression> パラメータには、21 以下の値を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

値 TH& の双曲正弦関数（sinh）、双曲余弦関数（cosh）の値を求めます。

```
SINH& = (Exp(TH&)-Exp(-TH&)) / 2
COSH& = (Exp(TH&) + Exp(-TH&)) / 2
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「Cos (P.8-59)」
「GetUnitData (P.8-138)」
「Log (P.8-174)」
「Sin (P.8-283)」
「UnitData (P.8-311)」

Fcopy

ファイルをコピーします。

・書式

Fcopy <srcPath>, <dstPath>

・パラメータ

パラメータ名	データ型	説明
<srcPath>	文字列型	コピー元ファイルの絶対パス名
<dstPath>	文字列型	コピー先ファイルの絶対パス名

・戻り値

なし

・解説

<srcPath> パラメータで指定したファイル名のファイルを、<dstPath> パラメータで指定したファイル名のファイルとしてコピーします。
<srcPath> パラメータと<dstPath> パラメータには、コピー元ファイルとコピー先ファイルのファイル名を絶対パスで指定してください。
コピー先ファイルがすでに存在する場合には上書きします。存在しない場合には新規作成します。
以下の場合にはファイルはコピーされません。

- コピー元ファイルが存在しない
- コピー先フォルダが存在しない
- 外部ストレージが装着されていない
- 外部ストレージの空き容量が足りない

<srcPath> パラメータまたは<dstPath> パラメータに、フォルダ名までしか指定しない場合は、Illegal function call のエラーが発生します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

フォルダ"E:¥" 以下のファイル名"1280-720.bmp" ファイルを、"F:¥" フォルダにコピーします。

```
Fcopy "E:¥1280-720.bmp", "F:¥1280-720.bmp"
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「Dskf (P.8-102)」
「Kill (P.8-160)」
「Rmdir (P.8-224)」
「Isfile (P.8-154)」
「Mkdir (P.8-188)」

FigureType

処理ユニットに設定可能な図形種類を取得します。

・書式

FigureType(<unitNo>, <figureNo>)

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	処理ユニット番号（0～（現在シーンの処理ユニット数 -1)）
<figureNo>	整数型	取得する図形番号（参照：「9-7 図形番号一覧 (P.9-70)」）

・戻り値

図形種別情報を返します。

・解説

<unitNo> パラメータで指定した処理ユニットの、<figureNo> パラメータで指定した図形の図形種別情報を取得します。
取得する図形種別情報は、図形種類の OR になります。例えば長方形（図形種別 8）と円（図形種別 32）に対応している場合 40 が返ります。
図形種別情報については、「9-6 図形データ一覧 (P.9-67)」を参照してください。
存在していないユニット番号を指定した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

処理ユニット番号 2 の図形番号 1 の図形が長方形に対応している事を確認し、領域を設定します。

```
Rem 対象図形に設定可能な図形種別を取得します
TYPE& = FigureType(2,1)

Rem 対象図形に矩形が設定可能な場合、図形を設定します
If (TYPE& AND 8) <> 0 Then
  Dim DATA&(5)
  DATA&(0) = 65560
  DATA&(1) = 8
  DATA&(2) = 0
  DATA&(3) = 0
  DATA&(4) = 639
  DATA&(5) = 479
  SetUnitFigure 2, 1, DATA&()
EndIf
```

・使用可能なモジュール

ユニット演算マクロ ／ ユニットマクロ

・対応バージョン

バージョン 6.40 以降

・関連項目

「GetUnitFigure (P.8-139)」　　「SetUnitFigure (P.8-276)」

Fix

数値の小数点以下を切り捨てた整数値を求めます。

・書式

Fix(<expression>)

・パラメータ

パラメータ名	データ型	説明
<expression>	実数型	小数点以下を切り捨てる数式

・戻り値

整数型の小数点以下を切り捨てた値を返します。

・解説

<expression> パラメータで指定した数式の小数点以下を切り捨てた整数値を求めます。
<expression> パラメータに負の値を指定した場合は、Int 関数が指定した負の値を超えない最大の負の整数を返すのに対して、Fix 関数は指定した負の値以上の最小の負の整数値を返します。たとえば、Int(-7.2) は-8、Fix(-7.2) は-7 をそれぞれ返します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

計測結果の実数型の値の小数点以下を切り捨てて整数値にします。

```
NUMBER1& = Fix(9.7)
NUMBER2& = Fix(-9.7)
NUMBER3& = Fix(-9.2)
```

結果は以下のとおりになります。

```
NUMBER1& = 9
NUMBER2& = -9
NUMBER3& = -9
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「Int (P.8-153)」
「GetUnitData (P.8-138)」
「UnitData (P.8-311)」

For To Step Next

For 文と Next 文の間の文を繰り返し実行します。

・書式

For <variable> = <startValue> To <endValue> [Step <increment>]
<statement>
Next [<variable>]

・パラメータ

パラメータ名	データ型	説明
<variable>	整数型	繰り返し処理のループカウンタ変数
<startValue>	整数型	ループカウンタ変数の初期値
<endValue>	整数型	繰り返し処理を終了するループカウンタ変数の値
<increment>	整数型	ループカウンタ変数の増分
<statement>	-	繰り返し実行するステートメント

・戻り値

なし

・解説

<variable> パラメータで指定したループカウンタ変数を、<startValue> パラメータで指定した初期値から<endValue> パラメータで指定した繰り返し処理を終了する値まで、<increment> パラメータで指定した増分だけ繰り返し処理のたびにループカウンタ変数の値に加算して<statement> パラメータで指定した For ブロックのステートメントを繰り返し実行します。
<increment> パラメータを省略した場合は、ループカウンタ変数の値が繰り返しのたびに 1 だけ増加します。
For ブロックのステートメントの中で、Exit For ステートメントを使用した場合は、繰り返し実行を途中で強制的に終了します。
For ブロックのステートメントの中で、Goto 関数や Gosub 関数を使用して Do ブロックの中から外に制御を移したり、外から中に制御を移したりした場合は、予期せぬ動作になる可能性があります。
For 文と Next 文のいずれかを記述しなかった場合は、記述の内容によって NEXT without FOR、FOR without NEXT、EXIT without FOR のいずれかのエラーが発生します。
For 文だけを記述して Next 文を記述しなかった場合は、NEXT without FOR のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

ユニットマクロの*MEASUREPROC サブルーチンの中で、処理ユニット番号 1 から 4 のエッジ位置処理ユニットのエッジ位置 X を取得して、処理ユニット番号 6 から 9 のサーチ処理ユニットの計測座標 X 上限値の値に順に設定します。

```
*MEASUREPROC

   Dim POS#(3)

   Rem 計測結果を取得します
   GetUnitData 1, 5, POS#(0)
   GetUnitData 2, 5, POS#(1)
   GetUnitData 3, 5, POS#(2)
   GetUnitData 4, 5, POS#(3)

   For NUM& = 0 To 3
      Rem 設定データを設定します
      SetUnitData NUM&+6, 136, POS#(NUM&)
   Next

Return
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「Do Loop While (P.8-71)」
「GetUnitData (P.8-138)」
「SetUnitData (P.8-275)」

GetAll

すべての入力端子の入力状態を取得します。

・書式

GetAll(<ioIdent>)

・パラメータ

パラメータ名	データ型	説明
<ioIdent>	文字列型	使用する通信モジュールの識別名 （常に"Parallello"） （参照：「9-4 IO モジュール一覧 (P.9-25)」）

・戻り値

整数型のすべての入力端子の入力状態の値を返します。
入力状態は、入力端子ごとの OFF(0) / ON(1) を 2 進数表記の各桁で表現した整数の値です。
パラレル I/O の場合は、DI0 から DI7 を 1 桁めから 8 桁めで表現した整数の値を返します。

- （例）DI0 から DI5 が ON、DI6～DI7 が OFF の場合
  2 進数表記：0011 1111
  取得できる入力状態の値：63

・解説

<ioIdent> パラメータで指定した通信モジュールの、すべての入力端子の入力状態を取得します。
<ioIdent> パラメータには、通常は"Parallello" を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

通信コマンドマクロで、パラレル I/O の DI 入力状態を取得します。

```
IOMODULE$ = "Parallello"

Rem 入力状態を取得します
STATE& = GetAll(IOMODULE$)
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「BusyOut (P.8-34)」
「GetPort (P.8-129)」
「JudgeOut (P.8-159)」
「PutAll (P.8-204)」
「PutPort (P.8-205)」
「RunOut (P.8-225)」

GetGlobalData

グローバルデータを取得します。

・書式

GetGlobalData <dataIdent>, <data>

・パラメータ

パラメータ名	データ型	説明
<dataIdent>	文字列型	取得するグローバルデータの識別名
<data>	整数型 実数型 文字列型	取得データ

・戻り値

なし

・解説

<dataIdent> パラメータで指定した識別名のグローバルデータの値を取得します。<data> パラメータには、取得したグローバルデータの値が指定した変数の型に変換されて格納されます。ただし、数値に変換できない文字列を取得した場合で<data> パラメータに整数型または実数型の変数が指定されている場合は、取得データに 0 が格納されます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
<dataIdent> パラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
<data> パラメータに定数を指定した場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

識別名が"ABC" のグローバルデータに設定された整数値を取得します。

```
Rem グローバルデータ"ABC"の値に整数値 1 を設定します。
SetGlobalData "ABC", 1

Rem グローバルデータ"ABC"の値を取得し、整数型の変数 DATA&に格納します。
GetGlobalData "ABC", DATA&
```

グローバルデータ予約語の"LayoutNoLocal_IoCommand" に格納されている現在のレイアウト番号の値を取得します。

```
Rem グローバルデータ"LayoutNoLocal_IoCommand"の値を取得します。
GetGlobalData "LayoutNoLocal_IoCommand", DATA&
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「AddGlobalData (P.8-23)」　　「SetGlobalData (P.8-253)」

GetImageSize

処理ユニットの画像の画像サイズを取得します。

・書式

GetImageSize <unitNo>, <measureImageNo>, <sizeX>, <sizeY>

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	処理ユニット番号（0～（現在シーンの処理ユニット数 -1)）
<measureImageNo>	整数型	取得する現在の処理対象ユニットの画像番号（参照：「9-9 画像番号一覧 (P.9-74)」、「9-10 サブ画像番号一覧 (P.9-77)」）
<sizeX>	整数型	取得した画像サイズ X
<sizeY>	整数型	取得した画像サイズ Y

・戻り値

なし

・解説

<unitNo> パラメータで指定した処理ユニットの持つ、<measureImageNo> パラメータで指定した画像番号の画像データのサイズを取得します。
<unitNo> パラメータに、カメラ画像入力（カメラ画像入力 FH、カメラ画像入力 HDR、カメラ画像入力 HDR Lite など）の処理項目を指定した場合は、Illegal function call のエラーが発生します。
<sizeX> パラメータと<sizeY> パラメータには、取得した画像サイズを格納する変数を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、1 回以上計測するか、または画像ファイルを指定して再計測した後、計測画像が表示された状態で使用してください。

・使用例

処理ユニット番号 2 の画像番号 0 の画像のサイズを取得します。

```
GetImageSize 2, 0, SIZEX&, SIZEY&
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「SaveMeasureImage (P.8-229)」　　「UnitNo (P.8-318)」
「Ut (P.8-320)」

GetImageWindow

画像表示ウィンドウの状態を取得します。

・書式

[シーン制御マクロ ／ 通信コマンドマクロ]

GetImageWindow <windowNo>, <locationX>, <locationY>, <width>, <height>, <unitNo>, <subNo>, <magnification>, <originX>, <originY>, <update>, <visible>

[ユニットマクロ]

GetImageWindow <locationX>, <locationY>, <width>, <height>, <unitNo>, <subNo>, <magnification>, <originX>, <originY>, <update>, <visible>

・パラメータ

パラメータ名	データ型	説明
<windowNo>	整数型	状態を取得する画像表示ウィンドウの番号
<locationX>	整数型	画像表示ウィンドウの左上 X 座標
<locationY>	整数型	画像表示ウィンドウの左上 Y 座標
<width>	整数型	画像表示ウィンドウの幅
<height>	整数型	画像表示ウィンドウの高さ
<unitNo>	整数型	表示対象の処理ユニットの処理ユニット番号
<subNo>	整数型	表示対象のサブ画像番号
<magnification>	実数型	表示倍率
<originX>	整数型	画像表示ウィンドウの左上座標に対する表示画像の左上 X 座標
<originY>	整数型	画像表示ウィンドウの左上座標に対する表示画像の左上 Y 座標
<update>	整数型	画像モード、画像ウィンドウの更新タイミング 0：計測時毎回（カメラ画像フリーズ） 1：計測時総合判定結果 NG の場合のみ（最新 NG 画像） 2：計測時対象処理ユニット NG の場合のみ 3：常時更新（カメラ画像スルー）
<visible>	整数型	表示有無 0：ウィンドウ非表示 1：ウィンドウ表示

・戻り値

なし

・解説

<windowNo> パラメータで指定した画像表示ウィンドウの状態を取得します。このマクロ関数をユニットマクロで使用する場合は、MEASUREDISPI サブルーチンで表示処理を実行した画像表示ウィンドウが対象になります。

<locationX> パラメータと<locationY> パラメータには、取得した画像コンテナウィンドウの左上座標から画像表示ウィンドウの左上座標への相対座標の値を格納する変数を指定してください。

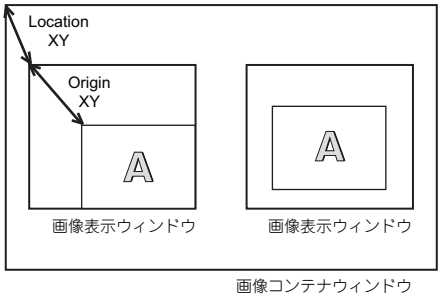
<width> パラメータと<height> パラメータには、取得した画像表示ウィンドウの幅と高さの値を格納する変数をそれぞれ指定してください。

<unitNo> パラメータには、取得した表示対象の処理ユニット番号の値を格納する変数を指定してください。画像表示ウィンドウの表示対象がフロー表示に連動する設定の場合は、-1 が格納されます。

<subNo> パラメータには、取得した表示対象のサブ画像番号の値を格納する変数を指定してください。画像表示ウィンドウの表示内容が位置一覧表示の場合は、-1 が格納されます。

<magnification> パラメータには、取得した画像表示ウィンドウの表示倍率の値を格納する変数を指定してください。表示倍率が自動の場合は、-1 が格納されます。

<originX> パラメータと<originY> パラメータには、取得した画像ウィンドウの左上座標から表示画像の左上座標への相対座標の値を格納する変数を指定してください。



<update> パラメータには、取得した画像表示ウィンドウの画像モードの値を格納する変数を指定してください。
<visible> パラメータには、取得した画像表示ウィンドウの表示状態の値を格納する変数を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

[ユニットマクロ]

- このマクロ関数は、*MEASUREDISPI サブルーチンまたは*MEASUREDISPG サブルーチンの中でしか使用できません。それ以外のサブルーチンで使用しようとした場合は、Illegal function call のエラーが発生します。

・使用例

通信コマンドマクロで、画像表示ウィンドウ 0～3 の画像モードをスルー表示に変更します。通信コマンドマクロでは、あらかじめ BusyOn フラグを ON の設定にしておきます。

```
For I& = 0 To 3
   Rem 画像表示ウィンドウの状態を取得します
   GetImageWindow I&, LOCATIONX&, LOCATIONY&, WIDTH&, HEIGHT&, UNITNO&, SUBNO&, MAG#, ORIGINX&, ORI
GINY&, UPDATE&, VISIBLE&

   Rem 更新タイミングをスルーに変更します
   UPDATE& = 3

   Rem 画像表示ウィンドウの状態を設定します
   SetImageWindow I&, LOCATIONX&, LOCATIONY&, WIDTH&, HEIGHT&, UNITNO&, SUBNO&, MAG#, ORIGINX&, ORI
GINY&, UPDATE&, VISIBLE&

Next
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「DisplayUnitNo (P.8-70)」
「SetDisplayUnitNo (P.8-249)」
「UnitNo (P.8-318)」
「GetTextWindow (P.8-137)」
「SetImageWindow (P.8-254)」
「Ut (P.8-320)」

GetMeasureOut

計測結果の外部出力設定を取得します。

・書式

GetMeasureOut

・パラメータ

なし

・戻り値

整数型の外部出力設定の値を返します。

- 0：外部に出力しない
- 1：外部に出力する

・解説

計測結果の外部出力設定として、レイアウト設定の「外部に出力する」の設定を取得します。(詳細は、『画像処理システム FH/FHV シリーズ ユーザーズマニュアル (SDNB-719)』の「レイアウトごとの出力信号の動作をアレンジする［レイアウト設定］」を参照してください。)
計測結果の外部出力設定の値が 0 の場合でも、マクロカスタマイズ機能で SendData 関数や SendString 関数を使用した場合はデータを出力します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。(詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。)

・使用例

シーン制御マクロで、「外部に出力する」が ON の場合は計測結果を外部装置に出力します。OFF の場合は計測結果をシステム状態監視コンソールウィンドウに出力します。

```
Rem 処理ユニット番号 3 の 2 次元コード処理ユニットの読み取り文字列を取得します
GetUnitData 3, "decodeCharStr", RESULT$

Rem 外部出力の設定内容で処理を分岐します
If GetMeasureOut = 1 Then
   SendString "TcpNormal", RESULT$
Else
   Print RESULT$
EndIf
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「SendData (P.8-246)」
「SendString (P.8-248)」
「SetMeasureOut (P.8-257)」

GetPlcData

ReadPlcMemory 関数で読み込んだデータを取り出します。

・書式

GetPlcData <ioIdent>, <readData()>, <offset>, <size>, <data>

・パラメータ

パラメータ名	データ型	説明
<ioIdent>	文字列型	使用する通信モジュールの識別名 （参照：「9-4 IO モジュール一覧 (P.9-25)」）
<readData()>	整数型配列	読み込んだデータ
<offset>	整数型	読み込んだデータの先頭からデータを取得するアドレスまでのオフセット（バイト単位）
<size>	整数型	取得するデータのサイズ（バイト単位）
<data>	整数型 実数型 文字列型	取得したデータ

・戻り値

なし

・解説

<ioIdent> パラメータで指定した通信モジュールを使用して、<readData()> パラメータで指定したデータ配列の先頭から<offset> パラメータで指定した値だけオフセットした位置の、<size> パラメータで指定したサイズ分のデータを取り出します。
ReadPlcMemory 関数を実行して PLC のメモリ領域の値を読み込んだ後、このマクロ関数でデータを取り出してください。
<readData()> パラメータには、ReadPlcMemory 関数で読み込んだデータを格納した 1 次元の整数型配列変数を、要素番号を指定せず() だけ付加して指定してください。
<offset> パラメータと<size> パラメータには、バイト単位のオフセットとサイズを指定してください。 ReadPlcMemory 関数で指定する単位（チャネル単位）とは異なります。
<size> パラメータには、2／4／8 のいずれかを指定してください。それぞれ、2 バイト整数／4 バイト整数／8 バイト実数として取得されます。
<data> パラメータには、取得したデータを格納する変数を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- PLC リンク通信で ReadPlcMemory 関数を使用して PLC のメモリ領域の値を読み込んだ場合は、必ずこのマクロ関数を使用して読み込んだデータから値を取り出してください。このマクロ関数を使用せずに ReadPlcMemory 関数のパラメータから直接値を取得する場合は、正しい値を取得できない可能性があります。

・使用例

通信コマンドマクロで、PLC リンクで接続された PLC から複数のデータを読み込みます。

```
IOMODULE$ = "UdpPlcLink"

Rem データ出力エリアの設定を取得します
GetSystemData IOMODULE$, "outputArea", AREA&
GetSystemData IOMODULE$, "outputMemoryAddress", ADDRESS&

Rem 読み込むデータを格納する整数型配列変数を作成します
Dim DATA&(1)

Rem データ出力エリアからデータ（4CH 分）を読み込みます
ReadPlcMemory IOMODULE$, AREA&, ADDRESS&, 4, DATA&()

Rem 読み込んだデータから値を取得します
GetPlcData IOMODULE$, DATA&(), 0, 4, VALUE0&
GetPlcData IOMODULE$, DATA&(), 4, 4, VALUE1&
```

・使用可能なモジュール
シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 4.20 以降

・関連項目

「ReadPlcMemory (P.8-210)」
「SetPlcData (P.8-258)」
「WritePlcMemory (P.8-329)」

GetPollingState

通信モジュールの実行状態を取得します。

・書式
GetPollingState(<ioIdent>)

・パラメータ

パラメータ名	データ型	説明
<ioIdent>	文字列型	実行状態を取得する通信モジュールの識別名（参照：「9-4 IO モジュール一覧 (P.9-25)」）

・戻り値
整数型の実行状態の値を返します。
- False：停止状態
- True：動作状態

・解説
<ioIdent> パラメータで指定した通信モジュールの実行状態を取得します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例

ユニットマクロ処理ユニットの*MEASUREPROC サブルーチンで、無手順 TCP 通信のデータを受信します。

```
Rem データを受信するために 12 バイト受信できるバッファを用意します
Dim BUFFER&(11)
IOMODULE$ = "TcpNormal"

Rem データを受信するために通信モジュールの実行状態を停止状態に設定します
SetPollingState IOMODULE$, False

Rem 受信データサイズを初期化します
SIZE& = 0

Rem データを受信するまで受信処理を繰り返します
Try
   Do
      Rem データの受信を試みます
      ReceiveData IOMODULE$, BUFFER&(), 12, SIZE&

      Rem データを受信したらデータサイズをシステム状態監視コンソールウィンドウに表示します
      If(SIZE& > 0) Then
         Print "Received data size = " + Str$(SIZE&)
      Endif
   Loop While SIZE& = 0

   Rem データを受信したので通信モジュールの実行状態を動作状態に設定します
   SetPollingState IOMODULE$, True

Catch
   Rem 停止状態にしている通信モジュールの実行状態を動作状態に戻します
   If GetPollingState(IOMODULE$) = False Then
      SetPollingState IOMODULE$, True
   Endif

End Try
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 4.20 以降

・関連項目

「SendData (P.8-246)」
「SendString (P.8-248)」
「SetPollingState (P.8-260)」
「ReceiveData (P.8-212)」

GetPort

指定した入力端子の入力状態を取得します。

・書式

GetPort(<ioIdent>, <portNo>)

・パラメータ

パラメータ名	データ型	説明
<ioIdent>	文字列型	使用する通信モジュールの識別名 （常に"Parallello"） （参照：「9-4 IO モジュール一覧 (P.9-25)」）
<portNo>	整数型	入力状態を取得する入力端子の端子番号 パラレル I/O の場合 • FH DI0～DI7：0～7 DSA N：100+N×8（N：ライン番号（0～7）） STEP N：101+N×8 DILINE0～DILINE2：200～202

・戻り値

整数型の入力端子の入力状態の値を返します。

- 0：入力 OFF 状態
- 1：入力 ON 状態

・解説

<ioIdent> パラメータで指定した通信モジュールの、<portNo> パラメータで指定した端子番号の入力端子の状態を取得します。
<ioIdent> パラメータには、通常は"Parallello" を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

通信コマンドマクロで、パラレル I/O の DI7 の入力状態を取得します。

```
IOMODULE$ = "Parallello"

Rem 入力状態を取得します
STATE& = GetPort(IOMODULE$, 7)
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「BusyOut (P.8-34)」
「GetPort (P.8-129)」
「JudgeOut (P.8-159)」
「PutAll (P.8-204)」
「PutPort (P.8-205)」
「RunOut (P.8-225)」

GetProfileData

設定ファイル（.ini ファイル）のデータを取得します。

・書式

GetProfileData <fileName>, <section>, <key>, <defaultData>, <data>
GetProfileData <fileName>, <section>, <key()>, <data()>

・パラメータ

パラメータ名	データ型	説明
<fileName>	文字列型	対象ファイルの絶対パス名
<section>	文字列型	対象のセクション名
<key>, <key()>	文字列型	対象のキー名
<defaultData>	文字列型	取得できなかった場合のデータ
<data>, <data0>	文字列型	取得したデータ

・戻り値

なし

・解説

<fileName>パラメータで指定したファイルの、<section>パラメータで指定したセクション名と<key>パラメータで指定したキー名に対応するデータを、<data>パラメータに格納します。
<fileName>パラメータとして、存在しないファイル名を指定した場合は、Illegal function call のエラーが発生します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

.ini ファイルの Setting セクション内のキー名 Logging のデータを取得して、システムデータに設定します。

```
GetProfileData "E:¥Setting.ini", "Setting", "Logging", "E:¥tmp", Logging$
SetSystemData "Logging","imageLoggingDirectory", Logging$
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 6.40 以降

・関連項目

「DeleteProfileData (P.8-65)」　「SetProfileData (P.8-262)」

GetSceneData

シーン制御マクロに関するデータやシーン変数の値を取得します。

・書式

GetSceneData<dataIdent>,<data>

・パラメータ

パラメータ名	データ型	説明
<dataIdent>	文字列型	取得するデータの識別名
<data>	整数型 実数型 文字列型	取得したデータ

・戻り値

なし

・解説

<dataIdent> パラメータには、シーン制御マクロのプログラムで使用している変数名または、シーン変数名を指定してください。シーン変数名は、TDM エディタのシーン変数登録画面で表示されている"SC." 以降の文字列です。
<dataIdent> パラメータで指定した識別名に対応したデータを取得します。
<data> パラメータには、取得したデータを格納する変数を指定してください。
取得対象のデータのデータ型と<data> パラメータで指定する変数のデータ型が異なる場合は、<data> パラメータのデータ型に変換した値が格納されます。
パラメータとして存在しない識別名を指定した場合は、Illegal function call のエラーが発生します。
パラメータとして、異なるデータ型の組み合わせを指定しても、エラーは発生しません。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。(詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。)

・使用例

シーン変数の値を任意の変数に代入します。

```
Rem シーン変数の SC.ABC& に格納されている値を、整数型の TMP& に代入します
GetSceneData "ABC&", TMP&

Rem シーン変数の SC.ASDF&() という配列変数のひとつめの配列要素の値を、
Rem 整数型の TMP& に代入します
GetSceneData "ASDF&(0)", TMP&
```

通信コマンドマクロで、シーン制御マクロ内で定義されている整数型の変数"SEARCHRESULT0&"、"SEARCHRESULT1&" を取得して計算に使用します。

```
Rem あらかじめシーン制御マクロで定義しておいた変数の値を取得します
GetSceneData "SEARCHRESULT0&", RET0&
GetSceneData "SEARCHRESULT1&", RET1&

Rem 取得した変数値を元に結果を算出します
RESULTDATA& = (RET0& + RET1&) / 2
```

・使用可能なモジュール

- シーン変数に対して適用する場合
  ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ
- シーン変数以外に対して適用する場合
  通信コマンドマクロ

・対応バージョン
バージョン 5.20 以降
（シーン変数に対しての適用は、バージョン 6.30 以降）

・関連項目

「SetSceneData (P.8-263)」

GetSceneGroupData

指定した識別名のシーングループデータの内容を取得します。

・書式
GetSceneGroupData <sceneGroupNo>, <dataIdent>, <data>

・パラメータ

パラメータ名	データ型	説明
<sceneGroupNo>	整数型	シーングループ番号
<dataIdent>	文字列型	シーングループデータの識別名 "sceneGroupTitle" : <data> で指定した文字列型変数にシーングループタイトルを取得します。シーングループタイトルは、言語識別子を使った多言語書式で指定できます。 言語識別子を使った記載の方法については、『画像処理システム FH/FHV シリーズ ユーザーズマニュアル (SDNB-719)』の「文字を入力する」を参照してください。
<data>	文字列型	取得したデータ

・戻り値
なし

・解説
<sceneGroupNo> パラメータで指定したシーングループに対して <dataIdent> パラメータで指定した識別名に対応したデータを取得します。<data> パラメータには取得したデータを格納する変数を指定してください。
パラメータとして存在しない識別名を指定した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。(詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。)

・使用例
現在のシーングループのタイトルシステム状態監視コンソールウィンドウに表示します。

```
MeasureStop
  GetSceneGroupData -1, "sceneGroupTitle", TITLE$
  Print LanguageString$(TITLE$)
MeasureStart
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン
バージョン 6.50 以降

・関連項目

「MeasureStart (P.8-182)」　「MeasureStop (P.8-183)」
「SetSceneGroupData (P.8-266)」

GetSystemData

システムデータまたは、システム変数の値を取得します。

・書式
GetSystemData <dataIdent0>, <dataIdent1>, <data>

・パラメータ

パラメータ名	データ型	説明
<dataIdent0>	文字列型	取得するシステムデータの識別情報 0 のデータ識別名
<dataIdent1>	文字列型	取得するシステムデータの識別情報 1 のデータ識別名
<data>	整数型 実数型 文字列型	取得したシステムデータの値

・戻り値
なし

・解説
- システムデータの値を取得する場合
  <dataIdent0> パラメータで指定した識別情報 0 に属する、<dataIdent1> パラメータで指定した識別情報 1 のシステムデータの値を取得します。<data> パラメータには、取得したシステムデータを格納する変数を指定してください。システムデータの識別情報については、「9-3 システムデータ一覧 (P.9-10)」をご覧ください。
- システム変数の値を取得する場合
  <dataIdent0> に"macro_sy" という文字列を設定し、<dataIdent1> に TDM エディタのシステム変数登録画面で表示されている"SY." 以降の変数名を設定します。 <data> パラメータには、取得したシステムデータを格納する変数を指定してください。

パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない識別名を指定した場合は、Illegal function call のエラーが発生します。
<dataIdent1> パラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
- バージョン 6.40 以降では、［クイックアクセス設定ツール］で登録されているため、RAMDisk および USBDisk の絶対パスは取得できません。これらの絶対パスを取得する場合は、AbsolutePath$ 関数を使用してください。
- <dataIdent0> パラメータに MultiLineRandom を指定する場合は、ライン 0 でないとシステムデータの値を正しく取得できません。

・使用例
システム変数の値を任意の変数に代入します。

```
Rem システム変数の SY.ABC& に格納されている値を、整数型の TMP& に代入します
GetSystemData "macro_sy", "ABC&", TMP&

Rem システム変数に SY.ASDF&() という配列変数のひとつめの配列要素の値を、
Rem 整数型の TMP& に代入します
GetSystemData "macro_sy", "ASDF&(0)", TMP&
```

識別情報 0 が"Measure" の計測制御設定に属する、識別情報 1 が"captureDirectory" の画面キャプチャ保存先フォルダの設定値を取得し、Sample.bmp ファイルを"E:¥temp¥bmp" にコピーします。

```
Rem 計測制御設定に属する画面キャプチャ保存先フォルダを取得します
GetSystemData "Measure", "captureDirectory", DIRNAME$

Rem コピー元のファイルの保存先パスを含むファイル名を求めます
FILE$ = DIRNAME$ + "¥Sample.bmp"

Rem ファイルをコピーします
Fcopy FILE$, "E:¥temp¥bmp¥Sample.bmp"
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降
（システム変数に対しての適用はバージョン 6.30 以降）

・関連項目

「AbsolutePath$ (P.8-22)」
「AddSystemData (P.8-24)」
「Fcopy (P.8-115)」
「GetGlobalData (P.8-121)」
「GetUnitData (P.8-138)」
「SetSystemData (P.8-271)」

GetText$

メッセージファイルからテキストデータを取得します。

・書式

GetText$(#<textDataNo>, <textIdent>)

・パラメータ

パラメータ名	データ型	説明
<textDataNo>	整数型	取得するテキストを含むメッセージファイルのテキストデータ番号（0～15）
<textIdent>	文字列型	取得するテキストデータの識別名

・戻り値

文字列型の取得したテキストデータの値を返します。

・解説

- メッセージファイルは、画面に表示したいメッセージを定義するファイルです。言語ごとに 1 ファイル作成されています。
- メッセージファイルのファイル名は、<メッセージファイル識別名>_<言語識別名>.msg で構成されています。
  <メッセージファイル識別名> は、OpenTextData 関数で<ident> として指定するメッセージファイルの識別名です。
  <言語識別名> は、言語を表す文字列です。

簡体中国語：chs
繁体中国語：cht
ドイツ語：deu
英語：eng
スペイン語：esp
フランス語：fra
イタリア語：ita
日本語：jpn
韓国語：kor

- メッセージファイルの書式は、1 行ごとに<textIdent> と指定する識別名と、それに対応するテキストデータを記述されています。
  <識別名 1> = <テキストデータ 1>
  <識別名 2> = <テキストデータ 2>
  <識別名 3> = <テキストデータ 3>
  ：
  <識別名 n> = <テキストデータ n>

<textDataNo> パラメータで指定したテキストデータ番号のメッセージファイルから、<textIdent> パラメータで指定した識別名のテキストデータを取得します。
<textDataNo> パラメータとして、OpenTextData 関数でメッセージファイルを開いたときに指定したテキストデータ番号を指定してください。
<textDataNo> パラメータとして、0～15 の範囲を超える値を指定した場合は、Illegal function call のエラーが発生します。
<textDataNo> パラメータとして、開かれていないテキストデータ番号を指定した場合は、Illegal function call のエラーが発生します。
<textIdent> パラメータとして、存在しない識別名を指定しても、エラーは発生しません。存在しない識別名を指定した場合は、「#ERROR」の文字列を返します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
ユニットマクロ処理ユニットの*MEASUREDISPT サブルーチンの中で、処理ユニット番号 5 のサーチ処理ユニットで計測した相関値を、処理ユニットのメッセージファイルから取得した文字列とともにテキスト表示ウィンドウに表示します。相関値は外部参照データ番号 5 です。

```
*MEASUREDISPT

    Rem 計測結果を取得します
    GetUnitData 5, 5, CR#

    Rem メッセージファイルを開きます
    OpenTextData "Search" As #1

    Rem テキストを取得します
    TEXT$ = GetText$(#1, "Correlation")

    Rem テキスト表示ウィンドウにメッセージファイルから取得した文字列を改行なしで描画します
    DrawText TEXT$, UnitJudge(5), 0

    Rem テキスト表示ウィンドウに計測結果を描画します
    DrawText Str2$(CR#, 4, 4, 0, 0), UnitJudge(5), 1

    Rem メッセージファイルを閉じます
    CloseTextData

Return
```

結果は以下のとおりになります。

```
相関値：90.0000
```

・使用可能なモジュール

ユニット演算マクロ ／ ユニットマクロ

・対応バージョン

バージョン 5.00 以降

・関連項目

「CloseTextData (P.8-46)」
「DrawText (P.8-97)」
「GetUnitData (P.8-138)」
「OpenTextData (P.8-196)」
「UnitJudge (P.8-317)」

GetTextWindow

テキスト表示ウィンドウの状態を取得します。

・書式

GetTextWindow <unitNo>, <subNo>, <update>, <visible>

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	表示対象の処理ユニットの処理ユニット番号
<subNo>	整数型	表示対象のサブ番号
<update>	整数型	更新タイミング（常に 0）
<visible>	整数型	表示有無 0：ウィンドウ非表示 1：ウィンドウ表示

・戻り値

なし

・解説

テキスト表示ウィンドウの状態を取得します。
<unitNo> パラメータには、取得した表示対象の処理ユニット番号の値を格納する変数を指定してください。テキスト表示ウィンドウの表示対象がフロー表示に連動する設定の場合は、-1 が格納されます。
<subNo> パラメータには、取得した表示対象のサブ画像番号の値を格納する変数を指定してください。
<update> パラメータには、取得した更新タイミングの値を格納する変数を指定してください。
<visible> パラメータには、取得したテキスト表示ウィンドウの表示状態を格納する変数を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

通信コマンドマクロで、テキスト表示ウィンドウで表示する処理ユニットの処理ユニット番号を、通信コマンドの引数で指定した番号に変更します。

```
Rem テキスト表示ウィンドウの状態を取得します
GetTextWindow UNITNO&, SUBNO&, UPDATE&, VISIBLE&

Rem 表示する処理ユニット番号にコマンドの引数で指定された番号を設定します
SetTextWindow argumentValue#(0), SUBNO&, UPDATE&, VISIBLE&
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「DisplayUnitNo (P.8-70)」
「SetDisplayUnitNo (P.8-249)」
「GetImageWindow (P.8-123)」
「SetTextWindow (P.8-274)」

「UnitNo (P.8-318)」　「Ut (P.8-320)」

GetUnitData

処理ユニットのデータを取得します。

・書式

GetUnitData <unitNo>, <dataNo>, <data>
GetUnitData <unitNo>, <dataIdent>, <data>

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	処理ユニット番号（0～（現在シーンの処理ユニット数 -1)）
<dataNo>	整数型	取得する処理ユニットデータの外部参照データ番号（参照：『画像処理システム FH/FHV シリーズ 処理項目機能リファレンスマニュアル（SDNB-713)』）
<dataIdent>	文字列型	取得する処理ユニットデータのデータ識別名
<data>	整数型 実数型 文字列型	取得した処理ユニットデータ

・戻り値

なし

・解説

<unitNo> パラメータで指定した処理ユニットの持つ、<dataNo> パラメータで指定した外部参照データ番号のデータを取得します。<data> パラメータには、取得したデータを格納する変数を指定してください。<dataNo> パラメータの代わりに<dataIdent> パラメータを指定してデータを取得することもできます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

処理ユニット番号 2 の処理ユニットの判定結果を取得します。判定結果は外部参照データ番号 0、外部参照データ識別名"JG" です。

```
GetUnitData 2, 0, JUDGE&

Rem 0 の代わりに"JG" を指定しても同じ結果が得られます
GetUnitData 2, "JG", JUDGE&
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「GetUnitFigure (P.8-139)」　「SetUnitData (P.8-275)」

「SetUnitFigure (P.8-276)」
「UnitData$ (P.8-312)」
「UnitNo (P.8-318)」
「UnitData (P.8-311)」
「UnitData2 (P.8-313)」
「Ut (P.8-320)」

GetUnitFigure

処理ユニットの図形データを取得します。

・書式

GetUnitFigure <unitNo>, <figureNo>, <figure()>

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	処理ユニット番号（0～（現在シーンの処理ユニット数 -1)）
<figureNo>	整数型	取得する図形データ（参照：「9-7 図形番号一覧 (P.9-70)」）
<figure()>	整数型配列	取得した図形データ（参照：「9-7 図形番号一覧 (P.9-70)」）

・戻り値

なし

・解説

<unitNo> パラメータで指定した処理ユニットの、<figureNo> パラメータで指定した図形の図形データを取得します。
<figure()> パラメータには、図形データを格納する 1 次元の整数型配列変数を、要素番号を指定せず () だけ付加して指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

処理ユニット番号 2 のエッジ処理ユニットの図形が幅付き円弧の場合、幅の値を変更します。

```
Dim FIGURE&(10)

Rem 処理ユニットの図形データを取得します
GetUnitFigure 2, 0, FIGURE&()

Rem 図形が幅付き円弧の場合
If FIGURE&(1) = 256 Then
   Rem 幅の値を設定します
   FIGURE&(7) = 64
   SetUnitFigure 2, 0, FIGURE&()
Endif
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「FigureType (P.8-116)」
「SetUnitData (P.8-275)」
「UnitData (P.8-311)」
「Ut (P.8-320)」
「GetUnitData (P.8-138)」
「SetUnitFigure (P.8-276)」
「UnitNo (P.8-318)」

Gosub

指定したサブルーチンを実行します。

・書式

Gosub <label>

・パラメータ

パラメータ名	データ型	説明
<label>	文字列型	実行するサブルーチンのラベル名

・戻り値

なし

・解説

<label> パラメータで指定したラベル名のサブルーチンの処理を実行します。
サブルーチンの処理の最後の Return 文で呼び出し元のサブルーチンに復帰して、このマクロ関数を実行した次の行から処理を再開します。
Return 文で呼び出し元のサブルーチンに復帰しない場合は、Goto 関数を使用してください。
パラメータとして存在しないラベルを指定した場合は、Undefined label のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

ユニットマクロの*MCRINIT サブルーチンで、別に定義した*INITPROC サブルーチンの処理を実行します。

```
*MCRINIT

   Rem 他のサブルーチンを実行します
   Gosub *INITPROC

Return

*INITPROC

   Dim DATA$(255)

Return
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「Goto (P.8-141)」

Goto

指定したラベルの行へ処理を移します。

・書式

Goto <label>

・パラメータ

パラメータ名	データ型	説明
<label>	整数型 文字列型	移動先の行番号またはラベル名

・戻り値

なし

・解説

<label> パラメータで指定した行番号の行またはラベル名の行に処理を移します。
Return 文で呼び出し元のサブルーチンに復帰する場合は、Gosub 関数を使用してください。
パラメータとして存在しないラベルを指定した場合は、Undefined label のエラーが発生します。
パラメータとして存在しない行番号を指定した場合は、Undefined line number のエラーが発生します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

ユニットマクロの*MEASUREPROC サブルーチンで、ラベル*PROC2 の行に処理を移してシステム状態監視コンソールウィンドウにテキストを出力します。最初の Goto 関数でラベル*PROC1 の行または*PROC3 の行に処理を移した場合は、何もせず終了します。最初の Goto 関数を実行しなかった場合も、ラベル*PROC3 の行に処理を移して何もせず終了します。

```
*MEASUREPROC

    Rem 他の行に処理を移します
    Goto *PROC2

    *PROC1
    Goto *PROC3

    *PROC2
    Print "PROC1"

    *PROC3

Return
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「Gosub (P.8-140)」

Hex$

数式の値を 16 進数表現の文字列に変換します。

・書式

Hex$(<expression>)

・パラメータ

パラメータ名	データ型	説明
<expression>	整数型	文字列に変換する数式

・戻り値

文字列型の 16 進数表現の値を返します。
負数の場合は、2 の補数で表現された 16 進数表現の値を返します。
値には 16 進数を表す(&H) は付加されません。

・解説

<expression> パラメータで指定した値を、16 進数表現の文字列に変換します。
<expression> パラメータに実数型の数式を指定する場合は、あらかじめ Fix 関数または Int 関数で整数型に変換した数式を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

ユニット演算マクロで、処理ユニット番号 5 番の OCR 処理ユニットの読取文字列"A" を取得して 16 進数に変換します。読取文字列は外部参照データ番号 20 です。

```
Rem 処理ユニットの計測結果を取得します
GetUnitData 5, 20, CHARA1$

Rem 文字をキャラクタコードの値に変換します
CODE& = Asc(CHARA1$)

Rem キャラクタコードの値を 16 進数表現の文字列に変換します
CHARA2$ = Hex$(CODE&)
```

結果は以下のとおりになります。

```
CHARA1$ = "A"
CODE& = 65
CHARA2$ = 41
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「Asc (P.8-31)」
「Chr$ (P.8-41)」
「GetUnitData (P.8-138)」
「LCase$ (P.8-162)」
「Left$ (P.8-163)」
「Len (P.8-164)」
「Mid$ (P.8-186)」
「Piece$ (P.8-200)」
「Right$ (P.8-223)」
「Str$ (P.8-287)」
「Str2$ (P.8-288)」
「UCase$ (P.8-309)」
「Val (P.8-321)」

If Then Else

指定した条件によって処理の流れを制御します。

・書式
If <expression> Then <statement>¦<label>[Else <statement>¦<label>]

・パラメータ

パラメータ名	データ型	説明
<expression>	-	処理の流れを制御する論理式（参照：「4-1-5 演算子 (P.4-12)」）
<statement>	-	処理するステートメント
<label>	文字列型	処理の移行先のラベル名

・戻り値
なし

・解説
<expression> パラメータで指定した条件が成り立つ場合に、<statement> パラメータで指定した If ブロックのステートメントを実行します。<statement> パラメータの代わりに<label> パラメータを指定した場合は、<label> パラメータで指定したラベルの行に処理を移します。<expression> パラメータで指定した条件が成り立たない場合は、Else 文の後の<statement> パラメータで指定した Else ブロックのステートメントを実行します。Else 文の後の<statement> パラメータの代わりに<label> パラメータを指定した場合は、<label> パラメータで指定したラベルの行に処理を移します。
Else 文を省略した場合は、<expression> パラメータで指定した条件が成り立つ場合だけ処理の流れを制御します。
Else 文は改行して記述することはできません。If 文から Else 文までを 1 行で記述してください。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
ユニットマクロの*MEASUREPROC サブルーチンの中で、処理ユニット番号 0 の処理ユニットの判定結果によって処理を分岐します。

```
*MEASUREPROC

   If UnitJudge(0)=JUDGE_OK Then Gosub *OKOUT Else Gosub *NGOUT

Return

*OKOUT
   Print "OK"
Return

*NGOUT
   Print "NG"
Return
```

・使用可能なモジュール
ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「Gosub (P.8-140)」
「Print (P.8-202)」
「Select Case Case Else End Select (P.8-244)」
「If Then Elseif Else EndIf (P.8-144)」
「UnitJudge (P.8-317)」

If Then Elseif Else EndIf

指定した条件によって処理の流れを制御します。

・書式
If <expression> Then
<ifStatement>
[Elseif <expression> Then
<elseifStatement>]
:
:
[Else
<elseStatement>]
EndIf

・パラメータ

パラメータ名	データ型	説明
<expression>	-	処理の流れを制御する論理式（参照：「4-1-5 演算子 (P.4-12)」）
<ifStatement>	-	If 文の後の論理式が真のときに実行するステートメント
<elseifStatement>	-	Elseif 文の後の論理式が真のときに実行するステートメント
<elseStatement>	-	すべての論理式が偽のときに実行するステートメント

・戻り値
なし

・解説
<expression> パラメータで指定した論理式が真の場合に、If ブロックまたは複数の Elseif ブロックの中から<statement> パラメータで指定したステートメントを実行します。

If 文の後の<expression> パラメータで指定した論理式が真の場合は、<ifStatement> パラメータで指定した If ブロックのステートメントを処理します。
If 文の後の<expression> パラメータで指定した論理式が偽の場合、かつ Elseif 文の後の<expression> パラメータで指定した論理式が真の場合は、<elseifStatement> パラメータで指定した Elseif ブロックのステートメントを処理します。
すべての<expression> パラメータで指定した論理式が偽の場合は、<elseStatement> パラメータで指定した Else ブロックのステートメントを処理します。
複数の<expression> パラメータで指定した論理式が真の場合は、If 文の論理式から順に最初に真になった論理式のブロック内のステートメントだけを実行します。
Elself ブロックと Else ブロックは省略可能です。
ステートメントの中で Goto 関数を使用して Do ブロックの中から外に制御を移したり、外から中に制御を移したりした場合は、予期せぬ動作になる可能性があります。
If 文と EndIf 文のいずれかまたは両方を記述しなかった場合は、記述の内容によって ELSEIF without IF、ELSE without IF、ENDIF without IF、IF without ENDIF、ELSEIF without ENDIF、ELSE without ENDIF のいずれかのエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
ユニットマクロの*MEASUREDISPG サブルーチンの中で、処理ユニット番号 1 のサーチ処理ユニットの相関値の値によってシステム状態監視コンソールウィンドウに出力するテキストを変更します。

```
*MEASUREDISPG

   Rem 計測結果を取得します
   GetUnitData 1, 5, RESULT&

   Rem 計測結果の値によって処理を分岐します
   If RESULT&>=80 Then
      DrawTextG "Excellent", 100, 100, 0
   Elseif RESULT&>=60 Then
      DrawTextG "Good", 100, 100, 0
   Else
      DrawTextG "Bad", 100, 100, 0
   EndIf

Return
```

・使用可能なモジュール
ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「DrawTextG (P.8-99)」
「If Then Else (P.8-143)」
「GetUnitData (P.8-138)」
「Select Case Case Else End Select (P.8-244)」

ImageFormat

処理ユニットの画像の画像フォーマットを取得します。

・書式

ImageFormat(<unitNo>, <measureImageNo>)

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	処理ユニット番号（0～（現在シーンの処理ユニット数 -1)）
<measureImageNo>	整数型	画像フォーマットを取得する計測画像番号（常に 0）

・戻り値

整数型のフォーマット形式の値を返します。

- 0：2 値画像
- 1：モノクロ画像
- 2：カラー画像
- -1：無効画像

・解説

<unitNo> パラメータで指定した処理ユニットの<measureImageNo> パラメータで指定した画像番号の画像のフォーマットを取得します。
<unitNo> パラメータに、カメラ画像入力（カメラ画像入力 FH、カメラ画像入力 HDR、カメラ画像入力 HDR Lite など）の処理項目を指定した場合は、Illegal function call のエラーが発生します。
<measureImageNo> パラメータには、通常は 0 を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

ユニットマクロの*MEASUREDISPT サブルーチンの中で、画像フォーマットに応じた文字列をテキスト表示ウィンドウに表示します。

```
*MEASUREDISPT

   Rem 処理ユニットの判定結果を取得します
   JUDGE& = UnitJudge(UnitNo)

   Rem 画像フォーマットを取得します
   FORMAT& = ImageFormat(UnitNo, 0)

   If FORMAT& = 2 Then
      Rem カラー画像の場合は"Color" と表示します
      DrawText "Color", JUDGE&, 1
   Elseif FORMAT& = 1 Then
      Rem モノクロ画像の場合は"Monocrome" と表示します
      DrawText "Monochrome", JUDGE&, 1
   EndIf
Return
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「DrawText (P.8-97)」
「UnitJudge (P.8-317)」
「UnitTitle$ (P.8-319)」
「Ut (P.8-320)」

ImageUpdate

カメラからの入力画像を更新します。

・書式

ImageUpdate

・パラメータ

なし

・戻り値

なし

・解説

計測フローの中にある画像入力関連の処理ユニットと画像変換関連の処理ユニットの画像を、カメラからの最新の画像を使用した画像に更新します。

・使用上の注意

なし

・使用例

通信コマンドマクロで、画像モードが「カメラ画像フリーズ」で表示された画像を最新の画像に更新します。

```
Rem フリーズ表示画像を最新の画像に更新します
ImageUpdate

Rem 更新した画像を画像表示ウィンドウの表示に反映します
RefreshImageWindow
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「GetImageWindow (P.8-123)」
「RefreshImageWindow (P.8-215)」
「SetImageWindow (P.8-254)」

Input#

ファイルからデータを読み込みます。

・書式

Input# <fileNo>, <data>[, <data>...]

・パラメータ

パラメータ名	データ型	説明
<fileNo>	整数型	読み込むファイルのファイル番号（0～15）
<data>	整数型 実数型 文字列型 配列型	読み込んだデータ

・戻り値

なし

・解説

<fileNo> パラメータで指定したファイル番号のファイルから、改行コードで区切られた 1 行の中に存在するカンマ区切りのデータを読み込みます。
<data> パラメータには、読み込んだデータを格納する変数を指定してください。
<data> パラメータには、読み込んだデータの値が指定した変数の型に変換されて格納されます。ただし、数値に変換できない文字列を読み込んだ場合で<data> パラメータに整数型または実数型の変数が指定されている場合は、読み込んだデータに 0 が格納されます。
読み込むデータがダブルクォーテーション（"）で囲まれた文字の場合は、ダブルクォーテーションも文字として読み込まれます。
読み込むデータの行の末尾に改行コードが存在しない場合は、Illegal function call のエラーが発生します。
<fileNo> パラメータとして、開かれていないファイル番号を指定した場合は、Illegal function call のエラーが発生します。
<fileNo> パラメータとして、Open For Input As# 関数以外のマクロ関数で開かれたファイルのファイル番号を指定した場合は、Illegal function call のエラーが発生します。
<fileNo> パラメータとして、0～15 の範囲を超える値を指定した場合は、Illegal function call のエラーが発生します。
<data> パラメータで指定した変数の数と読み込むデータの数が一致しない場合は、Illegal function call のエラーが発生します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

ファイルの終端に達するまで改行コードで区切られたデータを読み込みます。

```
Dim ALLDATA$(255)

Rem ファイルを開きます
Open "E:¥input.dat" For Input As #1

For I&=0 to 255
   Rem ファイル先頭から改行コードで区切られた行を 1 行ずつ読み込みます
   Input #1, DATA$
   ALLDATA$(I&) = DATA$

   Rem ファイルの終端に達したかどうかチェックします
   If Eof(1) <> 0 Then
      Exit For
   Endif

Next

Rem ファイルを閉じます
Close #1
```

・使用可能なモジュール
ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「Close (P.8-45)」
「Input$ (P.8-150)」
「Open For Input As# (P.8-193)」
「Eof (P.8-104)」
「Line Input# (P.8-165)」

Input$

ファイルからバイナリデータを読み込みます。

・書式
Input$(<length>[, #<fileNo>])

・パラメータ

パラメータ名	データ型	説明
<length>	整数型	読み込むデータのバイト数（0～255）
<fileNo>	整数型	読み込むファイルのファイル番号（0～15）

・戻り値
文字列型の読み込んだバイナリデータの値を返します。

・解説
<fileNo> で指定したファイル番号のファイルから、<length> パラメータで指定したサイズ分のバイナリデータを文字列として読み込みます。
<length> パラメータとして、255 バイトを超える値を指定した場合は、255 バイト目までのデータを読み込みます。
<length> パラメータで指定した読み込むデータのバイト数よりも、読み込むファイルに含まれるデータのバイト数が大きい場合は、<length> パラメータで指定したバイト数だけデータを読み込みます。読み込まれなかったデータは、このマクロ関数を次回実行したときに読み込みます。

読み込むデータがダブルクォーテーション（"）で囲まれた文字の場合は、ダブルクォーテーションも文字として読み込まれます。
<length> パラメータとして、読み込む全データのバイト数よりも大きな値を指定した場合は、Illegal function call のエラーが発生します。
<fileNo> パラメータとして、開かれていないファイル番号を指定した場合は、Illegal function call のエラーが発生します。
<fileNo> パラメータとして、Open For Input As# 関数以外のマクロ関数で開かれたファイルのファイル番号を指定した場合は、Illegal function call のエラーが発生します。
<fileNo> パラメータとして、0～15 の範囲を超える値を指定した場合は、Illegal function call のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
ファイルからバイナリデータを 6 バイト読み込みます。

```
Rem ファイルを開きます
Open "E:¥input.dat" For Input As #1

Rem ファイルからデータを 6 バイト読み込みます
DATA$ = Input$(6, #1)

Rem ファイルを閉じます
Close #1
```

・使用可能なモジュール
ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「Close (P.8-45)」
「Input$ (P.8-150)」
「Line Input# (P.8-165)」
「Open For Input As# (P.8-193)」

InsertUnit

処理ユニットを挿入します。

・書式

InsertUnit <unitNo>, <itemIdent>

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	処理ユニットを挿入する計測フロー上の処理ユニット番号（0～（現在シーンの処理ユニット数））
<itemIdent>	文字列型	挿入する処理ユニットの処理項目識別名「9-5 処理項目識別名一覧 (P.9-64)」

・戻り値

なし

・解説

<unitNo> パラメータで指定した計測フロー上の位置に、<itemIdent> パラメータで指定した識別名の処理項目を処理ユニットとして登録します。
<unitNo> パラメータで指定した処理ユニット番号以降の処理ユニットは、処理ユニット番号が 1 繰り下げられます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
<itemIdent> パラメータとして、存在しない識別名を指定した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）

・使用例

処理ユニット番号 2 と処理ユニット番号 3 の間に、サーチの処理ユニットを挿入します。

```
InsertUnit 3, "Search"
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「AssignUnit (P.8-32)」
「CopyUnit (P.8-54)」
「MeasureStart (P.8-182)」
「MoveUnit (P.8-190)」
「CheckUnit (P.8-40)」
「DeleteUnit (P.8-66)」
「MeasureStop (P.8-183)」
「UnitCount (P.8-310)」

Int

数値を整数値に変換します。

・書式

Int(<expression>)

・パラメータ

パラメータ名	データ型	説明
<expression>	実数型	整数値を求める数式

・戻り値

整数型の値を返します。

・解説

<expression> パラメータで指定した数式の小数点以下を切り捨て、指定した数式の値を超えない最大の整数の値に変換します。
<expression> パラメータに負の値を指定した場合は、Fix 関数が指定した負の値以上の最小の負の整数値を返すのに対して、Int 関数は指定した負の値を超えない最大の負の値を返します。たとえば、Fix(-7.2) は-7、Int(-7.2) は-8 をそれぞれ返します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

計測結果の実数型の値の小数点以下を切り捨てて整数値に変換します。

```
NUMBER1& = Int(9.7)
NUMBER2& = Int(-9.7)
NUMBER3& = Int(-9.2)
```

結果は以下のとおりになります。

```
NUMBER1& = 9
NUMBER2& = -10
NUMBER3& = -10
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「Fix (P.8-117)」
「GetUnitData (P.8-138)」
「UnitData (P.8-311)」

lsfile

ファイルの属性と有無を調べます。

・書式

lsfile(<fileName>)

・パラメータ

パラメータ名	データ型	説明
<fileName>	文字列型	調べるファイルの絶対パス名

・戻り値

整数型のファイルの属性と有無の値を返します。

- 0：ファイルなし
- 1：属性がファイル
- 2：属性がフォルダ

・解説

<fileName> パラメータで指定したファイル名のファイルの属性と有無を調べます。
<fileName> パラメータには、調べるファイルのファイル名を絶対パスで指定してください。
<fileName> パラメータで外部ストレージの中のファイルを指定する場合で外部ストレージがセンサコントローラに未装着の場合は、ファイルなしとして扱われます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定しても、エラーは発生しません。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

ファイルを開く前にファイルの有無を確認します。

```
Rem ファイルの有無を確認します
If lsfile("E:¥input.dat") <> 1 Then
   Print "ファイルがありません"
Endif
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「Dskf (P.8-102)」
「Fcopy (P.8-115)」
「GetSystemData (P.8-133)」
「Kill (P.8-160)」
「Open For Append As# (P.8-192)」
「Open For Input As# (P.8-193)」
「Open For Output As# (P.8-194)」
「Print (P.8-202)」

ItemCount

使用可能な処理項目の種類の数を取得します。

・書式
ItemCount

・パラメータ
なし

・戻り値
整数型の使用可能な処理項目の種類の数の値を返します。

・解説
センサコントローラ上で使用することができる処理項目の種類の数を取得します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
処理項目の種類の数を取得して繰り返し処理でサーチ処理項目を検索して、サーチ処理項目の処理項目種別を取得します。

```
Rem 使用可能な処理項目数を取得します
INUM& = ItemCount

Rem 処理項目数だけ、サーチの処理項目を検索します
For I&=0 To INUM&-1
   If ItemIdent$(I&) = "Search" Then
      Rem サーチ処理項目の処理項目種別を取得します
      FIGNAME& = ItemInfo(I&, 0)
   Endif
Next
```

・使用可能なモジュール
シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「ItemIdent$ (P.8-156)」
「ItemInfo (P.8-157)」
「ItemTitle$ (P.8-158)」

ItemIdent$

処理項目の識別名を取得します。

・書式

ItemIdent$(<itemNo>)

・パラメータ

パラメータ名	データ型	説明
<itemNo>	整数型	処理項目の識別名を取得する処理項目番号

・戻り値

文字列型の処理項目識別名の値を返します。

・解説

<itemNo> パラメータで指定した処理項目番号を持つ処理項目の識別名を取得します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定しても、Illegal function call のエラーは発生しません。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

処理項目の種類の数を取得して繰り返し処理でサーチ処理項目を検索して、サーチ処理項目の処理項目種別を取得します。

```
Rem 使用可能な処理項目数を取得します
INUM& = ItemCount

Rem 処理項目数だけ、サーチの処理項目を検索します
For I&=0 To INUM&-1
   If ItemIdent$(I&) = "Search" Then
      Rem サーチ処理項目の処理項目種別を取得します
      FIGNAME& = ItemInfo(I&, 0)
   Endif
Next
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「ItemCount (P.8-155)」
「ItemInfo (P.8-157)」
「ItemTitle$ (P.8-158)」

ItemInfo

処理項目の情報を取得します。

・書式

ItemInfo(<itemNo>, <kind>)

・パラメータ

パラメータ名	データ型	説明
<itemNo>	整数型	情報を取得する処理項目番号
<kind>	整数型	情報種別 0：処理項目種別 処理項目の種別を表す番号です。以下の値を取得できます。 • 0：検査・計測する（計測） • 1：画像を取り込む（画像入力） • 2：画像を補正する（画像補正） • 3：検査・計測を補助する（計測補助） • 4：処理を分岐する（分岐制御） • 5：結果を外部へ出力する（結果出力） • 6：結果を画面へ表示する（結果表示）

・戻り値

整数型の処理項目の情報の値を返します。情報が存在しない場合は、-1 を返します。

・解説

<itemNo> パラメータで指定した処理項目の、<kind> パラメータで指定した情報を取得します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定しても、Illegal function call のエラーは発生しません。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

処理項目の種類の数を取得して繰り返し処理でサーチ処理項目を検索して、サーチ処理項目の処理項目種別を取得します。

```
Rem 使用可能な処理項目数を取得します
INUM& = ItemCount

Rem 処理項目数だけ、サーチの処理項目を検索します
For I&=0 To INUM&-1
   If ItemIdent$(I&) = "Search" Then
      Rem サーチ処理項目の処理項目種別を取得します
      FIGNAME& = ItemInfo(I&, 0)
   Endif
Next
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「ItemCount (P.8-155)」
「ItemIdent$ (P.8-156)」
「ItemTitle$ (P.8-158)」
「UnitInfo (P.8-314)」

ItemTitle$

処理項目のタイトル名を取得します。

・書式

ItemTitle$(<itemNo>)

・パラメータ

パラメータ名	データ型	説明
<itemNo>	整数型	タイトル名を取得する処理項目番号

・戻り値

文字列型のタイトル名の値を返します。

・解説

<itemNo> パラメータで指定した処理項目のタイトル名を取得します。
タイトル名は、言語設定にしたがった言語で取得できます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定しても、Illegal function call のエラーは発生しません。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

処理項目の種類の数を取得して繰り返し処理で処理項目番号ごとの処理項目のタイトル名を取得します。

```
Rem 使用可能な処理項目数を取得します
INUM& = ItemCount

Rem 処理項目数だけ処理を繰り返します
For I&=0 To INUM&-1
   Rem 処理項目のタイトル名を取得します
   TITLE$ = ItemTitle$(I&)
Next
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「ItemCount (P.8-155)」
「ItemIdent$ (P.8-156)」
「ItemInfo (P.8-157)」

JudgeOut

総合判定信号の出力状態を設定します。

・書式

JudgeOut <ioIdent>, <state>

・パラメータ

パラメータ名	データ型	説明
<ioIdent>	文字列型	使用する通信モジュールの識別名 （"ParallelIo" または "EtherCAT"） （参照：「9-4 IO モジュール一覧 (P.9-25)」）
<state>	整数型	出力する総合判定結果 JUDGE_OK：総合判定 OK JUDGE_NG：総合判定 NG 実際の出力は、指定した総合判定結果と総合判定出力（Total Judgement または OR) 信号の出力極性の設定内容で出力されます。 "ParallelIO" 以外の通信モジュールを使用する場合の総合判定出力信号の出力極性は、NG 時 ON です

・戻り値

なし

・解説

<ioIdent> パラメータで指定した通信モジュールの OR 信号をはじめとした総合判定信号に、<state> パラメータで指定した出力状態を設定します。
<ioIdent> パラメータには、通常は"ParallelIo" または"EtherCAT" を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
<ioIdent> パラメータとして、存在しない識別名を指定した場合は、Illegal function call のエラーが発生します。
<state> パラメータとして、存在しない出力状態の値を指定しても、エラーは発生しません。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

通信コマンドマクロで、パラレル I/O の OR 信号を ON に設定します。

```
IOMODULE$ = "ParallelIo"

Rem 出力状態を設定します
JudgeOut IOMODULE$, 1
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「BusyOut (P.8-34)」
「GetPort (P.8-129)」
「PutPort (P.8-205)」
「TotalJudge (P.8-300)」
「GetAll (P.8-120)」
「PutAll (P.8-204)」
「RunOut (P.8-225)」

Kill

ファイルまたはフォルダを削除します。

・書式

Kill <fileName>

・パラメータ

パラメータ名	データ型	説明
<fileName>	文字列型	削除するファイルまたはフォルダの絶対パス名

・戻り値

なし

・解説

<fileName> パラメータで指定したパス名のファイルまたはフォルダを削除します。
<fileName> パラメータは、ドライブ名を含めた絶対パス名で指定してください。
<fileName> パラメータには、ワイルドカードとして*（文字列パターン）と?（1 文字）を使用することができます。
ワイルドカードでは次のような指定が可能です。

.	拡張子があるすべてのファイルを指定します
*	すべてのファイルを指定します
???.*	3 文字のファイル名で拡張子を持つファイルを指定します
???????	ファイル名と拡張子が 7 文字のファイル（"." 含む）を指定します
A*.*	A で始まって拡張子を持つファイルを指定します
A*A*A	A を 3 文字以上含むファイル（拡張子含む）を指定します
????*	4 文字以上のファイル名のファイル（拡張子含む）を指定します
????*.*	4 文字以上のファイル名を持ち、かつ拡張子を持つファイルを指定します
*.??	2 文字の拡張子を持つファイルを指定します

以下の場合にはファイルを削除しません。
- 指定したファイルが存在しない
- 外部ストレージが装着されていない

パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

ドライブ名"M:¥" 以下のファイル名"1280-720.bmp" のファイルを削除します。

```
Kill "M:¥1280-720.bmp"
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「Dskf (P.8-102)」
「Fcopy (P.8-115)」
「Isfile (P.8-154)」

LanguageString$

現在の言語設定にしたがった言語で文字列を取得します。

・書式

LanguageString$(<srcString>)

・パラメータ

パラメータ名	データ型	説明
<srclabel>	文字列型	文字列フォーマット（言語識別子を使った記載） 言語識別子を使った記載の方法については、『画像処理システム FH/FHV シリーズ ユーザーズマニュアル (SDNB-719)』の「文字を入力する」を参照してください。

・戻り値

文字列を返します。

・解説

<srclabel> パラメータとして指定した文字列フォーマット（言語識別子を使った記載）から、現在の言語設定に応じた文字列を取得します。
言語識別子を使った記載ではない文字列や空文字列が指定された場合は、指定された文字列をそのまま返します。
文字列フォーマット（言語識別子を使った記載）で現在の言語に対応する文字列が指定されていない場合は、英語に対応する文字列が返ります。
現在の言語に対応する文字列も英語に対応する文字列も含まれていない場合は空の文字列が返ります。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

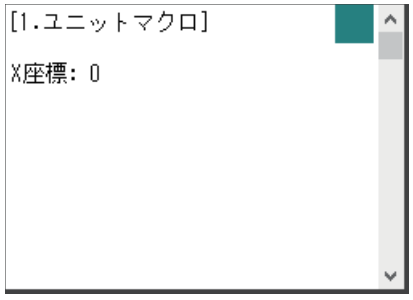
・使用上の注意

なし

・使用例

詳細結果表示に、現在の言語設定に応じた文字列を表示します。

```
*MEASUREDISPT
  TEXT$ = LanguageString$("¥jpn¥X 座標: ¥eng¥Position X: ")
  TEXT$ = TEXT$ + Str$(POSX#)
  DrawText TEXT$, UnitJudge(UnitNo), 0
Return
```



・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 5.75 以降

・関連項目
なし

LCase$

大文字を小文字に変換します。

・書式
LCase$(<string>)

・パラメータ

パラメータ名	データ型	説明
<string>	文字列型	小文字に変換するアルファベットを含む文字列

・戻り値
文字列型のアルファベット小文字に変換した値を返します。

・解説
<string> パラメータで指定した文字列に含まれるアルファベットの大文字を小文字に変換します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、256 文字め以降を切り捨てた 255 文字の文字列として扱われます。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
大文字を小文字に変換します。

```
CHARA1$ = "Measurement Result = 100.0(OK)"

Rem 文字列に含まれるアルファベットの大文字を小文字に変換します
CHARA2$ = LCase$(CHARA1$)
```

結果は以下のとおりになります。

```
CHARA2$ = "measurement result = 100.0(ok)"
```

・使用可能なモジュール
ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 4.20 以降

・関連項目

「Asc (P.8-31)」
「Chr$ (P.8-41)」
「Hex$ (P.8-142)」
「Left$ (P.8-163)」
「Len (P.8-164)」
「Mid$ (P.8-186)」

「Piece$ (P.8-200)」
「Right$ (P.8-223)」
「Str$ (P.8-287)」
「Str2$ (P.8-288)」
「UCase$ (P.8-309)」
「Val (P.8-321)」

Left$

文字列の左側から指定した長さの文字列を取り出します。

・書式

Left$(<string>, <length>)

・パラメータ

パラメータ名	データ型	説明
<string>	文字列型	取り出し元の文字列
<length>	整数型	取り出す文字列の長さ（1～取り出し元の文字列の長さ）

・戻り値

文字列型の取り出した値を返します。

・解説

<string> パラメータで指定した文字列の先頭である左側から、<length> パラメータで指定した長さの文字列を取り出します。
<length> パラメータには、取り出す文字数を文字のバイト数で指定してください。バイト数は、半角英数字や半角記号などの 1 バイト文字では 1 文字あたり 1 バイト、2 バイト文字では 1 文字あたり 2 バイトの値です。
<length> パラメータで指定した長さが<string> パラメータで指定した文字列より長い場合は、<string> パラメータで指定した文字列すべてを取り出します。
<length> パラメータとして、0 以下の値を指定した場合は、Illegal function call のエラーが発生します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、256 文字め以降を切り捨てた 255 文字の文字列として扱われます。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

文字列の先頭から 11 バイト分の文字列を取り出します。半角英字は 1 文字あたり 1 バイトなので、11 文字分の文字列を取り出すことができます。

```
CHARA$ = "Measurement Result"

Rem 文字列の先頭から半角英字 11 文字分の文字列を取り出します
TITLE$ = Left$(CHARA$, 11)
```

結果は以下のとおりになります。

```
TITLE$ = "Measurement"
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「Asc (P.8-31)」
「Chr$ (P.8-41)」
「Hex$ (P.8-142)」
「LCase$ (P.8-162)」
「Len (P.8-164)」
「Mid$ (P.8-186)」
「Piece$ (P.8-200)」
「Right$ (P.8-223)」
「Str$ (P.8-287)」
「Str2$ (P.8-288)」
「UCase$ (P.8-309)」
「Val (P.8-321)」

Len

指定した文字列の長さを求めます。

・書式
Len(<string>)

・パラメータ

パラメータ名	データ型	説明
<string>	文字列型	長さを求める文字列

・戻り値
整数型の文字列の長さのバイト数の値を返します。

・解説
<string> パラメータで指定した文字列の長さをバイト数で求めます。
バイト数は、半角英数字や半角記号などの 1 バイト文字では 1 文字あたり 1 バイト、2 バイト文字では 1 文字あたり 2 バイトの値です。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定しても、エラーは発生しません。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
文字列"OMRON" の長さを求めます。

```
CHRLEN& = Len(˝OMRON˝)
```

結果は以下のとおりになります。

```
CHRLEN& = 5
```

・使用可能なモジュール
ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「Asc (P.8-31)」
「Chr$ (P.8-41)」
「Hex$ (P.8-142)」
「LCase$ (P.8-162)」
「Left$ (P.8-163)」
「Mid$ (P.8-186)」
「Piece$ (P.8-200)」
「Right$ (P.8-223)」
「Str$ (P.8-287)」
「Str2$ (P.8-288)」
「UCase$ (P.8-309)」
「Val (P.8-321)」

Line Input#

ファイルから 1 行分のデータを読み込みます。

・書式

Line Input #<fileNo>, <data>

・パラメータ

パラメータ名	データ型	説明
<fileNo>	整数型	読み込むファイルのファイル番号（0～15）
<data>	文字列型	読み込んだデータ

・戻り値

なし

・解説

<fileNo> パラメータで指定したファイル番号のファイルから、改行コードで区切られた 1 行分のデータを読み込みます。Input# 関数が 1 行に含まれる複数のデータについてそれぞれデータ型を指定して読み込むのに対して、このマクロ関数は 1 行分のすべてのデータをひとつの文字列型のデータとして読み込みます。
<data> パラメータには、読み込んだデータを格納する変数を指定してください。
読み込む行が改行コードのみの行の場合は、読み込んだデータにヌル文字列("") が格納されます。
読み込むデータがダブルクォーテーション（"）で囲まれた文字の場合は、ダブルクォーテーションも文字として読み込まれます。
<fileNo> パラメータとして、開かれていないファイル番号を指定した場合は、Illegal function call のエラーが発生します。
<fileNo> パラメータとして、Open For Input As# 関数以外のマクロ関数で開かれたファイルのファイル番号を指定した場合は、Illegal function call のエラーが発生します。
<fileNo> パラメータとして、0～15 の範囲を超える値を指定した場合は、Illegal function call のエラーが発生します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

ファイルの終端に達するまでデータを読み込みます。

```
Dim ALLDATA$(255)

Rem ファイルを開きます
Open "E:¥input.dat" For Input As #1

For I&=0 to 255
   Rem ファイル先頭から1 行ずつ読み込みます
   Line Input #1, DATA$
   ALLDATA$(I&) = DATA$

   Rem ファイルの終端に達したかどうかチェックします
   If Eof(1) <> 0 Then
      Exit For
   Endif
Next

Rem ファイルを閉じます
Close #1
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「Close (P.8-45)」
「Input# (P.8-149)」
「Open For Input As# (P.8-193)」
「Eof (P.8-104)」
「Input$ (P.8-150)」

List

システム状態監視コンソールウィンドウにプログラム内容の全部または一部を出力します。

・書式

List [<lineNo1>][-<lineNo2>]

・パラメータ

パラメータ名	データ型	説明
<lineNo1>	整数型	出力を開始する行の行番号
<lineNo2>	整数型	出力を終了する行の行番号

・戻り値

なし

・解説

<lineNo1> パラメータで指定された行番号のプログラム行から、<lineNo2> パラメータで指定された行番号のプログラム行までのプログラムを、システム状態監視コンソールウィンドウに出力します。
<lineNo1> パラメータと<lineNo2> パラメータを省略した場合は、プログラムのすべての行が出力されます。<lineNo1> パラメータだけを省略した場合は、プログラムの先頭から<lineNo2> パラメータで指定された行番号のプログラム行までのプログラムが出力されます。
<lineNo2> パラメータだけを省略した場合は、<lineNo1> パラメータで指定された行番号のプログラム行だけが出力されます。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、Debug 関数でデバッグモードを指定している場合にのみ有効です。それ以外の場合は、Rem 関数と同様に、プログラム行の本関数の記述は無視されます。（詳細は、「6-1 デバッグ機能の使い方について (P.6-2)」を参照してください。

・使用例

ユニットマクロ処理ユニットで、Stop 関数でプログラムの実行を中断したあと、システム状態監視コンソールウィンドウで List 関数を実行して、プログラム行番号 230 から 240 までのプログラムをシステム状態監視コンソールウィンドウに出力します。

```
Macro(U1) Stop in 220
Macro(U1) List 230-240
230 POS.X#=(POS0.X@ + POS1.X@) / 2
240 POS.Y#=(POS0.Y@ + POS1.Y@) / 2
Macro(U1)>
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「Cls (P.8-48)」
「Cont (P.8-49)」
「Debug (P.8-63)」
「DebugPrint (P.8-64)」
「Print (P.8-202)」
「SetStop (P.8-270)」
「SetVar (P.8-281)」
「Stop (P.8-286)」

「SubList (P.8-292)」　　「VarList (P.8-322)」

LoadBackupData

システム＋シーングループ 0 データを読み込みます。

・書式

LoadBackupData(<fileName>)

・パラメータ

パラメータ名	データ型	説明
<fileName>	文字列型	読み込むシステム＋シーングループ 0 データファイル名（*.bkd）

・戻り値

なし

・解説

<fileName> パラメータで指定したシステム＋シーングループ 0 データファイルを読み込みます。
<fileName> パラメータには、読み込むファイルのファイル名を絶対パスで指定してください。
<fileName> パラメータには、システム＋シーングループ 0 データファイル名を指定してください。
システム＋シーングループ 0 データファイル以外のファイルを指定した場合は、Illegal function call のエラーが発生します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）
- ファイルを読み込んだ後は［本体に保存］を実行してください。その後読み込んだ設定をセンサコントローラに反映させるために、センサコントローラを再起動してください。

・使用例

システム＋シーングループ 0 データを読み込んだ後、［本体に保存］を実行します。その後読み込んだ設定をセンサコントローラに反映させるために、センサコントローラを再起動します。

```
Rem システム＋シーングループ 0 データファイルを読み込みます
LoadBackupData "E:¥BACKDIR¥BackupData.bkd"

Rem 本体に保存します
SaveData

Rem センサコントローラを再起動します
SystemReset
```

・使用可能なモジュール

通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「GetSystemData (P.8-133)」
「LoadSceneGroup (P.8-170)」
「LoadUnitData (P.8-172)」
「SaveData (P.8-227)」
「LoadScene (P.8-169)」
「LoadSystemData (P.8-171)」
「SaveBackupData (P.8-226)」
「SystemReset (P.8-294)」

LoadScene

シーンデータを読み込みます。

・書式

LoadScene <sceneNo>, <fileName>

・パラメータ

パラメータ名	データ型	説明
<sceneNo>	整数型	読み込み先のシーン番号（0～127）
<fileName>	文字列型	読み込むシーンデータファイル名（*.scn）

・戻り値

なし

・解説

<sceneNo> パラメータで指定したシーン番号に、<fileName> パラメータで指定したシーンデータファイルを読み込みます。
<fileName> パラメータには、読み込むファイルのファイル名を絶対パスで指定してください。
<fileName> パラメータには、シーンデータファイル名を指定してください。シーンデータファイル以外のファイルを指定した場合は、Illegal function call のエラーが発生します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）

・使用例

シーン 2 にシーンデータを読み込んだ後、使用するシーンをシーン 2 に切り替えます。

```
Rem シーンデータファイルを読み込みます
LoadScene 2, "E:¥BACKDIR¥scene02.scn"

Rem シーンを切り替えます
ChangeScene 2
```

・使用可能なモジュール

通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「ChangeScene (P.8-38)」
「LoadBackupData (P.8-168)」
「LoadSystemData (P.8-171)」
「SaveScene (P.8-231)」
「GetSystemData (P.8-133)」
「LoadSceneGroup (P.8-170)」
「LoadUnitData (P.8-172)」
「SceneNo (P.8-241)」

LoadSceneGroup

シーングループデータを読み込みます。

・書式

LoadSceneGroup <sceneGroupNo>, <fileName>

・パラメータ

パラメータ名	データ型	説明
<sceneGroupNo>	整数型	読み込み先のシーングループ番号（0～31）
<fileName>	文字列型	読み込むシーングループデータファイル名（*.sgp）

・戻り値

なし

・解説

<sceneGroupNo> パラメータで指定したシーングループ番号に、<fileName> パラメータで指定したシーングループデータファイルを読み込みます。
<fileName> パラメータには、読み込むファイルのファイル名を絶対パスで指定してください。
<fileName> パラメータには、シーングループデータファイル名を指定してください。シーングループデータファイル以外のファイルを指定した場合は、Illegal function call のエラーが発生します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）

・使用例

シーングループ 2 にシーンデータを読み込んだ後、使用するシーンをシーン 0 に切り替えます。

```
Rem シーングループデータファイルを読み込みます
LoadSceneGroup 2, "E:¥BACKDIR¥scenegroup02.sgp"

Rem シーンを切り替えます
ChangeScene 0
```

・使用可能なモジュール

通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「ChangeScene (P.8-38)」
「GetSystemData (P.8-133)」
「LoadBackupData (P.8-168)」
「LoadScene (P.8-169)」
「LoadSystemData (P.8-171)」
「LoadUnitData (P.8-172)」
「SaveSceneGroup (P.8-232)」
「SceneGroupNo (P.8-238)」

LoadSystemData

システムデータを読み込みます。

・書式

LoadSystemData <fileName>

・パラメータ

パラメータ名	データ型	説明
<fileName>	文字列型	読み込むシステムデータファイル名（*.ini）

・戻り値

なし

・解説

<fileName> パラメータで指定したシステムデータファイルを読み込みます。
<fileName> パラメータには、読み込むファイルのファイル名を絶対パスで指定してください。
<fileName> パラメータには、システムデータファイル名を指定してください。システムデータファイル以外のファイルを指定した場合は、Illegal function call のエラーが発生します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）
- ファイルを読み込んだ後は［本体に保存］を実行してください。その後読み込んだ設定をセンサコントローラに反映させるために、センサコントローラを再起動してください。

・使用例

システムデータを読み込んだ後、［本体に保存］を実行します。その後読み込んだ設定をセンサコントローラに反映させるために、センサコントローラを再起動します。

```
Rem システムデータファイルを読み込みます
LoadSystemData "E:¥BACKDIR¥backupsysset.ini"

Rem 本体に保存します
SaveData

Rem センサコントローラを再起動します
SystemReset
```

・使用可能なモジュール

通信コマンドマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「GetSystemData (P.8-133)」
「LoadScene (P.8-169)」
「LoadUnitData (P.8-172)」
「SaveSystemData (P.8-233)」
「LoadBackupData (P.8-168)」
「LoadSceneGroup (P.8-170)」
「SaveData (P.8-227)」
「SystemReset (P.8-294)」

LoadUnitData

処理ユニットデータを読み込みます。

・書式
LoadUnitData <sceneNo>, <unitNo>, <unitCount>, <mode>, <fileName>

・パラメータ

パラメータ名	データ型	説明
<sceneNo>	整数型	読み込み先のシーン番号（-1～127）
<unitNo>	整数型	読み込みを開始する処理ユニット番号（0～（現在シーンの処理ユニット数 -1））
<unitCount>	整数型	読み込む処理ユニット数（-1、1～（現在シーンの処理ユニット数 - 読み込みを開始する処理ユニット番号））
<mode>	整数型	読み込みモード 0：読み込み先の処理ユニットを上書き 1：読み込み先の処理ユニットの位置に挿入
<fileName>	文字列型	読み込む処理ユニットデータファイル名（*.unt）

・戻り値
なし

・解説
<sceneNo> パラメータで指定したシーン番号の、<unitNo> パラメータで指定した処理ユニット番号の処理ユニットから<unitCount> パラメータで指定した処理ユニット数だけ、<mode> パラメータで指定した読み込みモードで、<fileName> パラメータで指定した処理ユニットデータファイルを読み込みます。
<sceneNo> パラメータに-1 を指定した場合は、読み込み先のシーン番号には現在のシーンのシーン番号が指定されます。
<unitCount> パラメータに-1 を指定した場合は、処理ユニットデータファイルに含まれるすべての処理ユニットデータを読み込みます。
<mode> パラメータで読み込みモードを指定します。フォルダ、並列化、条件実行（If） など、複数の処理項目を組み合わせた処理ユニットに対して上書きモードでユニット読み込みを行わないでください。 ファイルに保存されたユニットと差し替えたい場合は、DeleteUnit 関数でフォルダを削除した上で挿入モードでファイルに保存されたユニットを読み込むようにしてください。
<fileName> パラメータには、読み込むファイルのファイル名を絶対パスで指定してください。
<fileName>パラメータには、処理ユニットデータファイル名を指定してください。処理ユニットデータファイル以外のファイルを指定した場合は、Illegal function call のエラーが発生します。また、このとき読み込み先のシーンデータはクリアされます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。

文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。(詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。)

・使用例

シーン 2 の処理ユニット番号 3 と処理ユニット番号 4 の間に、処理ユニットデータファイルを読み込んで同じ処理ユニットを 5 個挿入します。

```
Rem シーンデータファイルを読み込みます
LoadUnitData 2, 4, 5, 1, "E:¥BACKDIR¥unitsave.unt"
```

・使用可能なモジュール

通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「GetSystemData (P.8-133)」
「LoadScene (P.8-169)」
「LoadSystemData (P.8-171)」
「SceneNo (P.8-241)」
「Ut (P.8-320)」
「LoadBackupData (P.8-168)」
「LoadSceneGroup (P.8-170)」
「SaveUnitData (P.8-234)」
「UnitNo (P.8-318)」

Log

自然対数値を求めます。

・書式

Log(<expression>)

・パラメータ

パラメータ名	データ型	説明
<expression>	整数型 実数型	自然対数を求める数式

・戻り値

実数型の自然対数の値を返します。

・解説

<expression> パラメータで指定した数式の自然対数の値を求めます。
<expression> パラメータには、正の数を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

変数 X& に代入した整数 25 の自然対数を求めます。

```
X& = 25
XLOG# = Log(X&)
```

結果は以下のとおりになります。

```
XLOG# = 3.21887582487
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「Exp (P.8-114)」
「GetUnitData (P.8-138)」
「UnitData (P.8-311)」

Lsqumeth

複数点の座標から最小二乗法を使って近似直線を求めます。

・書式

Lsqumeth <count>, <x()>, <y()>, <line()>

・パラメータ

パラメータ名	データ型	説明
<count>	整数型	近似直線を求める座標の個数
<x>	実数型配列	近似直線を求める点の X 座標を格納した 1 次元配列
<y>	実数型配列	近似直線を求める点の Y 座標を格納した 1 次元配列
<line()>	実数型配列	求めた近似直線の直線成分

・戻り値

なし

・解説

<count> パラメータで指定した個数の、<x()> パラメータと<y()> パラメータで指定した座標をもとに近似直線を求めます。
<line()> パラメータには、求めた近似直線の直線成分を格納する変数を、要素番号を指定せず() だけ付加して指定してください。<line()> パラメータには、直線の方程式 ax+by+c=0 を満たす a,b,c のパラメータに関して、実数型配列変数の配列要素 0 に a、1 に b、2 に c がそれぞれ格納されます。
<count> パラメータには、2 以上の整数の値を指定してください。
<x()> パラメータと<y()> パラメータには、<count> パラメータで指定した個数以上の座標値を格納した 1 次元の実数型配列変数を、要素番号を指定せず() だけ付加して指定してください。
このマクロ関数は、おもに複数のエッジ計測などで求めた計測点から、その辺の直線を求めるときに使用します。
パラメータとして、定義されていない配列を指定した場合は、Undefined array のエラーが発生します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

4 つのエッジ点からワークの 1 辺の直線を求めます。

```
Dim POSX#(3), POSY#(3), PARAM#(2)

Rem 計測結果からエッジ点を取得します
For I&=0 To 3
   GetUnitData I&+1, "X", POSX#(I&)
   GetUnitData I&+1, "Y", POSY#(I&)
Next

Rem 直線成分を求めます
Lsqumeth 4, POSX#(), POSY#(), PARAM#()

Erase POSX#(), POSY#(), PARAM#(), DIST#()
```

・使用可能なモジュール
ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「ApproximationCircle (P.8-28)」
「Crspoint (P.8-60)」
「Dposline (P.8-72)」
「Erase (P.8-105)」
「GetUnitData (P.8-138)」

Measure

計測処理を実行します。

・書式
Measure[<wait>]

・パラメータ

パラメータ名	データ型	説明
<wait>	整数型	マクロ関数の復帰タイミング 0：計測終了を待たずに以降のプログラム行を実行します。 1：計測終了を待ってから以降のプログラム行を実行します。 2：計測終了と計測結果表示の終了を待ってから以降のプログラム行を実行します。 3：計測終了、計測結果表示の終了、UI 表示更新の終了を待ってから以降のプログラム行を実行します。

・戻り値
なし

・解説
<wait> パラメータで指定した復帰タイミングで計測処理を実行します。
<wait> パラメータを省略した場合は、<wait> パラメータに 0 を設定したときと同じ動作になります。
<wait> パラメータとして 0 を指定した場合は、このマクロ関数の実行直後に実行した計測処理が終了していない可能性があるため、計測結果を正しく取得できません。計測結果を取得したい場合は、<wait> パラメータに 1、2、3 のいずれかを指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定しても、Illegal function call のエラーは発生しません。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
- このマクロ関数は、BUSY 信号などの計測中信号が OFF した計測許可状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）

・使用例
通信コマンドマクロで、計測処理を実行した後に処理ユニット番号 2 のサーチ処理ユニットの計測 X 座標と計測 Y 座標を取得します。計測 X 座標は外部参照データ番号 6、計測 Y 座標は外部参照データ番号 7 です。

```
Rem 計測を実行して計測終了まで待ちます
Measure 1

Rem 計測結果を取得します
GetUnitData 2, 6, POSX#
GetUnitData 2, 7, POSY#
```

・使用可能なモジュール
通信コマンドマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「MeasureStart (P.8-182)」　「MeasureStop (P.8-183)」
「Remeasure (P.8-220)」

MeasureDispG

処理ユニットの計測結果の表示処理を実行します。

・書式
MeasureDispG <unitNo>, <subNo>

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	処理ユニット番号（0～（現在シーンの処理ユニット数 -1))
<subNo>	整数型	表示先の画像表示ウィンドウの画像表示サブ番号(参照:「9-10 サブ画像番号一覧 (P.9-77)」)

・戻り値
なし

・解説
<unitNo> パラメータで指定した処理ユニットの、<subNo> パラメータで指定された画像表示サブ番号の画像にグラフィックを表示します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
<unitNo> パラメータとして、存在しない番号、数値、データ型の組合わせ、値の組合わせを指定した場合は、Illegal function call のエラーが発生します。
<subNo> パラメータとして、存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定しても、エラーは発生しません。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
- このマクロ関数は、*MEASUREDISPI サブルーチンまたは*MEASUREDISPG サブルーチンの中でしか使用できません。それ以外のサブルーチンで使用しようとした場合は、Illegal function call のエラーが発生します。

・使用例
ユニットマクロ処理ユニットの*MEASUREDISPG サブルーチンの中で、処理ユニット番号 2 のサブ番号 0 の計測結果表示処理（グラフィック表示）を実行します。

```
*MEASUREDISPG
   MeasureDispG 2, 0
Return
```

・使用可能なモジュール
ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「UnitNo (P.8-318)」　　「Ut (P.8-320)」

MeasureDispI

処理ユニットのサブ画像番号の画像を表示します

・書式
MeasureDispI <unitNo>, <subNo>

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	処理ユニット番号（0～（現在シーンの処理ユニット数 -1））
<subNo>	整数型	処理項目の画像表示サブ番号（参照：「9-10 サブ画像番号一覧 (P.9-77)」）

・戻り値
なし

・解説
<unitNo> パラメータで指定した処理ユニットの、<subNo> パラメータで指定された画像表示サブ番号の画像にグラフィックを表示します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
<unitNo> パラメータとして、存在しない番号、数値、データ型の組合わせ、値の組合わせを指定した場合は、Illegal function call のエラーが発生します。
<subNo> パラメータとして、存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定しても、エラーは発生しません。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
- このマクロ関数は、*MEASUREDISPI サブルーチンまたは*MEASUREDISPG サブルーチンの中でしか使用できません。それ以外のサブルーチンで使用しようとした場合は、Illegal function call のエラーが発生します。

・使用例
ユニットマクロ処理ユニットの*MEASUREDISPI サブルーチンの中で、処理ユニット番号 0 の照度差ステレオ画像入力処理項目の形状 2 フィルタ画像を、ユニットマクロの出力画像として表示します。

```
*MEASUREDISPI
   MeasureDispI 0, 2
Return
```

・使用可能なモジュール
ユニットマクロ

・対応バージョン
バージョン 6.40 以降

・関連項目

「UnitNo (P.8-318)」　　「Ut (P.8-320)」

MeasureDispT

処理ユニットの詳細結果を表示します。

・書式
MeasureDispT <unitNo>, <subNo>

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	処理ユニット番号（0～（現在シーンの処理ユニット数 -1)）
<subNo>	整数型	処理項目の画像表示サブ番号（参照：「9-10 サブ画像番号一覧 (P.9-77)」）

・戻り値
なし

・解説
<unitNo> パラメータで指定した処理ユニットの、<subNo> パラメータで指定された画像表示サブ番号の画像にグラフィックを表示します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
<unitNo> パラメータとして、存在しない番号、数値、データ型の組合わせ、値の組合わせを指定した場合は、Illegal function call のエラーが発生します。
<subNo> パラメータとして、存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定しても、エラーは発生しません。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
- このマクロ関数は、*MEASUREDISPT サブルーチンの中でしか使用できません。それ以外のサブルーチンで使用しようとした場合は、Illegal function call のエラーが発生します。

・使用例
ユニットマクロ処理ユニットの*MEASUREDISPT サブルーチンの中で、処理ユニット番号 2 の演算処理項目の詳細結果を、ユニットマクロの詳細結果に表示します。

```
*MEASUREDISPT
   MeasureDispT 2, 0
Return
```

・使用可能なモジュール
ユニットマクロ

・対応バージョン

バージョン 6.40 以降

・関連項目

「UnitNo (P.8-318)」　　「Ut (P.8-320)」

MeasureId$

計測 ID を取得します。

・書式

MeasureId$

・パラメータ

なし

・戻り値

文字列型の計測 ID の値を返します。
計測 ID は センサコントローラが計測トリガを受け付けた時の日時情報にライン番号を合わせた値です。
計測 ID のフォーマット：YYYY-MM-DD_HH-MM-SS-XXXN
YYYY: 西暦、MM: 月、DD: 日、HH: 時、MM: 分、SS: 秒、XXX: ミリ秒、N: ライン番号になります。
例
計測時間：2007 年 12 月 24 日 11 時 10 分 25.500 秒で、ライン 0 の場合、計測 ID は「2007-12-24_11-10-25-5000」となります。

・解説

計測 ID を取得します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、*MEASUREPROC サブルーチンの中でしか使用できません。それ以外のサブルーチンで使用しようとした場合は、マクロ関数の処理は実行されずエラーも発生しません。

・使用例

ユニットマクロ処理ユニットの*MEASUREPROC サブルーチンの中で、計測 ID を取得して計測 ID をファイル名として処理ユニット計測画像を保存します。

```
*MEASUREPROC

    Rem 計測 ID を取得してファイル名の文字列を求めます
    FILENAME$ = MeasureId$ + ".bmp"

    Rem 計測画像 0 をビットマップフォーマットで出力します
    SaveMeasureImage 0, FILENAME$, 0

Return
```

・使用可能なモジュール

ユニット演算マクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「SaveMeasureImage (P.8-229)」

MeasureProc

処理ユニットの計測処理を実行します。

・書式

MeasureProc <unitNo>

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	処理ユニット番号（0～（現在シーンの処理ユニット数 -1)）

・戻り値

なし

・解説

<unitNo> パラメータで指定した処理ユニットの計測処理を実行します。
<unitNo> パラメータとして、画像入力関連の処理ユニットを指定した場合は、正しく処理できない場合があります。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

処理ユニット番号 2 のサーチ処理ユニットの計測処理を実行します。

```
MeasureProc 2
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 5.40 以降

・関連項目

「UnitNo (P.8-318)」　「Ut (P.8-320)」

MeasureStart

計測トリガの入力を許可します。

・書式

MeasureStart

・パラメータ

なし

・戻り値

なし

・解説

計測トリガの入力や通信コマンドの入力を許可して計測許可状態にします。
MeasureStop 関数で計測許可状態から計測禁止状態にした場合に、このマクロ関数で計測許可状態に戻すことができます。
計測トリガの入力を許可した場合は、BUSY 信号などの計測中信号が OFF します。

・使用上の注意

- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。(詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。)
- 計測禁止状態で実行しないといけないマクロ関数を実行する場合は、MeasureStop 関数で計測禁止状態にしてから実行しないと、予期せぬ動作になる場合があります。
  計測許可状態と計測禁止状態で、実行できるマクロ関数と実行できないマクロ関数があります。詳細は各マクロ関数の解説と使用上の注意をご覧ください。

・使用例

通信コマンドマクロで、計測トリガの入力を許可してから、処理ユニット番号 1 の形状サーチⅢ処理ユニットの計測処理を実行します。
計測処理が終わったら、計測トリガの入力を禁止して、計測結果を取得します。

```
Rem 計測トリガの入力を許可します
MeasureStart

Rem 計測処理を実行します
Measure 1

Rem 計測トリガの入力を禁止します
MeasureStop

Rem 形状サーチⅢ処理ユニットの計測結果を取得します
GetUnitData 1, 6, POSX#
GetUnitData 1, 7, POSY#
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「DisableUnit (P.8-68)」
「GetUnitData (P.8-138)」
「MeasureStop (P.8-183)」
「SetUnitData (P.8-275)」
「GetSceneGroupData (P.8-132)」
「Measure (P.8-176)」
「SetSceneGroupData (P.8-266)」

MeasureStop

計測トリガの入力を禁止します。

・書式

MeasureStop [<mode>]

・パラメータ

パラメータ名	データ型	説明
<mode>	整数型	計測トリガ入力禁止モード（常に 0）

・戻り値

なし

・解説

計測トリガの入力や通信コマンドの入力を禁止して計測禁止状態にします。計測終了と計測結果表示の終了を待ってから、計測禁止状態になります。
MeasureStart 関数を実行することで、計測禁止状態から計測許可状態に戻すことができます。
<mode> パラメータを省略した場合は、<mode> パラメータに 0 を指定した場合と同じ動作になります。
計測トリガの入力を禁止した場合は、BUSY 信号などの計測中信号が ON します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定しても、Illegal function call のエラーは発生しません。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、BUSY 信号などの計測中信号が OFF した計測許可状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）
- 計測禁止状態で実行しないといけないマクロ関数を実行する場合は、MeasureStop 関数で計測禁止状態にしてから実行しないと、予期せぬ動作になる場合があります。
  計測許可状態と計測禁止状態で、実行できるマクロ関数と実行できないマクロ関数があります。詳細は各マクロ関数の解説と使用上の注意をご覧ください。

・使用例

通信コマンドマクロで、計測トリガの入力を許可してから、処理ユニット番号 1 の形状サーチⅢ処理ユニットの計測処理を実行します。
計測処理が終わったら、計測トリガの入力を禁止して、計測結果を取得します。

```
Rem 計測トリガの入力を許可します
MeasureStart

Rem 計測処理を実行します
Measure 1

Rem 計測トリガの入力を禁止します
MeasureStop

Rem 形状サーチⅢ処理ユニットの計測結果を取得します
GetUnitData 1, 6, POSX#
GetUnitData 1, 7, POSY#
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「DisableUnit (P.8-68)」
「GetUnitData (P.8-138)」
「MeasureStart (P.8-182)」
「SetUnitData (P.8-275)」
「GetSceneGroupData (P.8-132)」
「Measure (P.8-176)」
「SetSceneGroupData (P.8-266)」

MemoryInfo

現在のメモリの使用状況（サイズ）を取得します。

・書式
MemoryInfo (<kind>)

・パラメータ

パラメータ名	データ型	説明
<kind>	整数型	取得情報種別 0: OS が認識したメモリ全体サイズ(MB) 1: OS が認識したメモリ使用可能サイズ(MB) 2: 拡張メモリ全体サイズ(MB) 3: 拡張メモリ使用可能サイズ(MB) 4: CoreRA プロセスメモリ全体サイズ(MB) 5: CoreRA プロセスメモリ使用可能サイズ(MB) 6: 処理ユニットデータ全体サイズ(MB) 7: 処理ユニットデータ使用可能サイズ（MB 10000: 物理メモリ全体サイズ（MB 10001: 物理メモリ使用可能サイズ（MB 10002: ページファイル全体サイズ（MB 10003: ページファイル使用可能サイズ（MB 10004: 仮想メモリ全体サイズ（MB 10005: 仮想メモリ使用可能サイズ（MB） 10010: 処理ユニットデータインデックス全体（メモリブロック数） 10011: 処理ユニットデータインデックス使用済(メモリブロック数） 10012: 処理ユニットデータインデックス使用可能(メモリブロック数） 10014: 処理ユニットデータ使用済みサイズ（MB） 10020: 画像データインデックス全体（メモリブロック数） 10021: 画像データインデックス使用済み（メモリブロック数） 10022: 画像データインデックス使用可能（メモリブロック数） 10024: 画像データ使用済みサイズ（MB） 10100: UWF オーバーレイ全体サイズ（MB） 10101: UWF オーバーレイ使用可能サイズ（MB）

・戻り値
取得したメモリのサイズ（整数値）を返します。

・解説
<kind> パラメータで指定した種別の情報を取得します。定義されていない <kind> を指定してもエラーは発生しません。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
物理メモリに一定量の空きがあるかをチェックします。

```
If MemoryInfo(5) > 100000000 Then
   Rem 物理メモリに一定量の空きがある場合の処理を記述
EndIf
```

・使用可能なモジュール
ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 6.41 以降

・関連項目
なし

MessageBox

メッセージボックスを表示します。

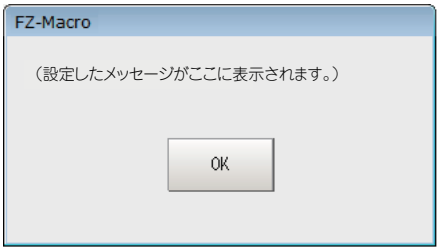
・書式
MessageBox <message>

・パラメータ

パラメータ名	データ型	説明
<message>	文字列型	メッセージボックス内に表示するメッセージの文字列

・戻り値
なし

・解説
表示されるメッセージボックスは、次のようになります。



パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定しても、Illegal function call のエラーは発生しません。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
ユニットマクロで関数を実行した場合、メッセージボックスが最前面に表示されないときがあります。最前面に表示されないときは、センサコントローラに USB キーボードを接続して、［Alt］＋［Tab］で表示する画面を切り替えてください。

・使用例
計測が NG の時に NG 表示を出して計測を止めます。

```
Rem 計測が NG の時に MessageBox を出して計測を止める

If totaljudge = -1 then
   MessageBox "NG"
Endif
```

・使用可能なモジュール
ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 5.40 以降

・関連項目

「If Then Elseif Else EndIf (P.8-144)」

Mid$

文字列の一部を取り出します。

・書式
Mid$(<string>, <start>, <length>)

・パラメータ

パラメータ名	データ型	説明
<string>	文字列型	取り出し元の文字列
<start>	整数型	取り出しを開始する位置（1～取り出し元の文字列の長さ）
<length>	整数型	取り出す文字列の長さ（1～取り出しを開始する位置から右側の文字列の長さ）

・戻り値
文字列型の取り出した値を返します。

・解説
<string> パラメータで指定した文字列の、<start> パラメータで指定した位置から、<length> パラメータで指定した長さの文字列を取り出します。
<length> パラメータには、取り出す文字数を文字のバイト数で指定してください。バイト数は、半角英数字や半角記号などの 1 バイト文字では 1 文字あたり 1 バイト、2 バイト文字では 1 文字あたり 2 バイトの値です。
<length> パラメータで指定した長さが、<start> パラメータで指定した位置から右側の文字列の長さより長い場合は、<string> パラメータで指定した文字列の<start> パラメータで指定した位置から右側の文字列すべてを取り出します。
<start> パラメータとして、取り出し元の文字列の長さより大きな値を指定した場合は、Illegal function call のエラーが発生します。
<start> パラメータまたは<length> パラメータとして、0 以下の値を指定した場合は、Illegal function call のエラーが発生します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、256 文字め以降を切り捨てた 255 文字の文字列として扱われます。

戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
半角英数字の文字列の 2 文字目から 4 文字分の文字列を取り出します。また、3 文字目から 8 文字分の文字列を取り出します。ただし、取り出し元の文字列は 8 文字のため、実際に取り出されるのは 3 文字目から末尾までの 5 文字です。

```
INPUTSTR$ = "ABCDEFG"

Rem 文字列の 2 文字目から 4 文字分の文字列を取り出します
OUTPUTSTR1$ = Mid$(INPUTSTR$, 2, 4)
Rem 文字列の 3 文字目から 8 文字分の文字列を取り出します
OUTPUTSTR2$ = Mid$(INPUTSTR$, 3, 8)
```

結果は以下のとおりになります。

```
OUTPUTSTR1$ = "BCDE"
OUTPUTSTR2$ = "CDEFG"
```

・使用可能なモジュール
ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「Asc (P.8-31)」
「Chr$ (P.8-41)」
「Hex$ (P.8-142)」
「LCase$ (P.8-162)」
「Left$ (P.8-163)」
「Len (P.8-164)」
「Piece$ (P.8-200)」
「Right$ (P.8-223)」
「Str$ (P.8-287)」
「Str2$ (P.8-288)」
「UCase$ (P.8-309)」
「Val (P.8-321)」

Mkdir

フォルダを作成します。

・書式

Mkdir <directoryName>

・パラメータ

パラメータ名	データ型	説明
<directoryName>	文字列型	作成するフォルダのフォルダ名

・戻り値

なし

・解説

<directoryName> パラメータで指定したフォルダを作成します。
<directoryName> パラメータには、作成するフォルダのフォルダ名を絶対パスで指定してください。
次のような場合は、フォルダを作成せず Illegal function call のエラーが発生します。

- 指定したフォルダがすでに存在する場合
- フォルダの作成先に外部ストレージを指定した場合で、外部ストレージがセンサコントローラに未装着の場合
- フォルダ作成先のドライブの空き容量が不足している場合

パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

E ドライブのルートに「IMAGE2」フォルダを作成します。

```
Mkdir "E:¥IMAGE2"
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「Dskf (P.8-102)」
「Fcopy (P.8-115)」
「GetSystemData (P.8-133)」
「Isfile (P.8-154)」
「Kill (P.8-160)」
「Rmdir (P.8-224)」

MOD

剰余を求めます。

・書式

<expression1> MOD <expression2>

・パラメータ

パラメータ名	データ型	説明
<expression1>	整数型	剰余を求める被除数の数式
<expression2>	整数型	剰余を求める除数の数式

・戻り値

整数型の剰余の値を返します。

・解説

<expression1> パラメータで指定した数式を、<expression2> パラメータで指定した数式で割った剰余を求めます。
<expression1> パラメータと<expression2> パラメータに実数型の数式を指定した場合は、小数部を四捨五入した値として扱われます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
<expression2> パラメータに 0 を指定した場合は、Division by zero のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

0～100 の間でカウンタをまわします。

```
I& = (I&+1) MOD 100
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「Fix (P.8-117)」
「GetUnitData (P.8-138)」
「UnitData (P.8-311)」

MoveUnit

処理ユニットを移動します。

・書式

MoveUnit <srcUnitNo>, <destUnitNo>

・パラメータ

パラメータ名	データ型	説明
<srcUnitNo>	整数型	移動元の処理ユニット番号（0～（現在シーンの処理ユニット数 -1)）
<destUnitNo>	整数型	計測フロー上の移動先の位置を示す処理ユニット番号（0～（現在シーンの処理ユニット数 -1)）

・戻り値

なし

・解説

<srcUnitNo> パラメータで指定した処理ユニットを、<destUnitNo> パラメータで指定した計測フロー上の位置に移動します。
<srcUnitNo> パラメータで指定した処理ユニット番号以降の処理ユニットは、処理ユニット番号が 1 繰り上げられます。<destUnitNo> パラメータで指定した処理ユニット番号以降の処理ユニットは、処理ユニット番号が 1 繰り下げられます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）

・使用例

処理ユニット番号 2 の処理ユニットを処理ユニット番号 5 と処理ユニット番号 6 の間に移動します。

```
MoveUnit 2, 5
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「AssignUnit (P.8-32)」
「CopyUnit (P.8-54)」
「InsertUnit (P.8-152)」
「MeasureStop (P.8-183)」
「CheckUnit (P.8-40)」
「DeleteUnit (P.8-66)」
「MeasureStart (P.8-182)」
「UnitCount (P.8-310)」

NOT

式の否定結果を求めます。

・書式

NOT(<expression>)

・パラメータ

パラメータ名	データ型	説明
<expression>	整数型	否定演算する数式

・戻り値

整数型の否定の値を返します。

・解説

<expression> パラメータで指定した数式の値を 32 桁の 2 進数として、各ビットを反転した値を否定結果として求めます。
<expression> パラメータに実数型の数式を指定した場合は、小数部を四捨五入した値として扱われます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

変数 X& に代入した整数 0 の否定を求めます。

```
X& = 0
XX& = NOT(X&)
```

結果は以下のとおりになります。

```
XX& = -1
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「AND (P.8-25)」
「GetUnitData (P.8-138)」
「UnitData (P.8-311)」
「OR (P.8-199)」
「XOR (P.8-331)」

Open For Append As#

ファイルを追加書き込みモードで開きます。

・書式

Open <fileName> For Append As #<fileNo>

・パラメータ

パラメータ名	データ型	説明
<fileName>	文字列型	開くファイルの絶対パス名
<fileNo>	整数型	開くファイルに割り当てるファイル番号（0～15）

・戻り値

なし

・解説

<fileName> パラメータで指定したファイルを、<fileNo> パラメータで指定したファイル番号を割り当てて追加書き込みモードで開きます。
このマクロ関数を実行して開いたファイルは、必ず Close 関数を実行して同じ処理の中で閉じてください。Close 関数を実行しない場合やこのマクロ関数を実行したサブルーチンとは別のサブルーチンや実行タイミングで実行した場合は、ファイルへの書き込みやファイルからの読み込みなどのファイルへのアクセスが正しく終了しない可能性があります。
<fileName> パラメータには、開くファイルのファイル名を絶対パスで指定してください。
<fileName> パラメータで指定したファイル名のファイルが存在しない場合は、ファイルを新規作成します。
<fileName>パラメータで指定したファイル名のファイルがすでに存在する場合は、存在するファイルに追加で書き込みができる状態でファイルを開きます。
<fileNo> パラメータとして、0～15 の範囲を超える値を指定した場合は、Illegal function call のエラーが発生します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

E ドライブにあるファイル「input.txt」を開いてデータを追加で書き込みます。

```
STRING$ = "Sample"

Rem ファイルを開きます
Open "E:¥input.txt" For Append As #1

Print #1, STRING$

Rem ファイルを閉じます
Close #1
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「Close (P.8-45)」
「GetSystemData (P.8-133)」
「Input$ (P.8-150)」
「Line Input# (P.8-165)」
「Open For Output As# (P.8-194)」
「Eof (P.8-104)」
「Input# (P.8-149)」
「Isfile (P.8-154)」
「Open For Input As# (P.8-193)」
「Print# (P.8-203)」

Open For Input As#

ファイルを読み取りモードで開きます。

・書式

Open <fileName> For Input As #<fileNo>

・パラメータ

パラメータ名	データ型	説明
<fileName>	文字列型	開くファイルの絶対パス名
<fileNo>	整数型	開くファイルに割り当てるファイル番号（0～15）

・戻り値

なし

・解説

<fileName> パラメータで指定したファイルを、<fileNo> パラメータで指定したファイル番号を割り当てて読み取りモードで開きます。
このマクロ関数を実行して開いたファイルは、必ず Close 関数を実行して同じ処理の中で閉じてください。Close 関数を実行しない場合やこのマクロ関数を実行したサブルーチンとは別のサブルーチンや実行タイミングで実行した場合は、ファイルへの書き込みやファイルからの読み込みなどのファイルへのアクセスが正しく終了しない可能性があります。
<fileName> パラメータには、開くファイルのファイル名を絶対パスで指定してください。
<fileName> パラメータで指定したファイル名のファイルが存在しない場合は、Illegal function call のエラーが発生します。
<fileNo> パラメータとして、0～15 の範囲を超える値を指定した場合は、Illegal function call のエラーが発生します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

ファイルの終端に達するまでデータを読み込みます。

```
Dim ALLDATA$(255)

Rem ファイルを開きます
Open "E:¥input.dat" For Input As #1

For I&=0 to 255
   Rem ファイル先頭から 1 行ずつ読み込みます
   Input #1, DATA$
   ALLDATA$(I&) = DATA$

   Rem ファイルの終端に達したかどうかチェックします
   If Eof(1) <> 0 Then
      Exit For
   Endif
Next

Rem ファイルを閉じます
Close #1
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「Close (P.8-45)」
「GetSystemData (P.8-133)」
「Input$ (P.8-150)」
「Line Input# (P.8-165)」
「Open For Output As# (P.8-194)」
「Eof (P.8-104)」
「Input# (P.8-149)」
「Isfile (P.8-154)」
「Open For Append As# (P.8-192)」
「Print# (P.8-203)」

Open For Output As#

ファイルを書き込みモードで開きます。

・書式

Open <fileName> For Output As #<fileNo>

・パラメータ

パラメータ名	データ型	説明
<fileName>	文字列型	開くファイルの絶対パス名
<fileNo>	整数型	開くファイルに割り当てるファイル番号（0～15）

・戻り値

なし

・解説

<fileName> パラメータで指定したファイルを、<fileNo> パラメータで指定したファイル番号を割り当てて書き込みモードで開きます。
このマクロ関数を実行して開いたファイルは、必ず Close 関数を実行して同じ処理の中で閉じてください。Close 関数を実行しない場合やこのマクロ関数を実行したサブルーチンとは別のサブルーチンや実行タイミングで実行した場合は、ファイルへの書き込みやファイルからの読み込みなどのファイルへのアクセスが正しく終了しない可能性があります。
<fileName> パラメータには、開くファイルのファイル名を絶対パスで指定してください。

<fileName> パラメータで指定したファイル名のファイルが存在しない場合は、指定したファイル名のファイルを新規作成して書き込みができる状態でファイルを開きます。
<fileName> パラメータで指定したファイル名のファイルがすでに存在する場合は、存在するファイルのデータをクリアしたうえで書き込みができる状態でファイルを開きます。
<fileNo> パラメータとして、0～15 の範囲を超える値を指定した場合は、Illegal function call のエラーが発生します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
ファイルを開いてデータを書き込んだ後、ファイルを閉じます。

```
DATA& = 10

Rem ファイルを開きます
Open "E:¥input.dat" For Output As #1

Rem 開いたファイルにデータを書き込みます
Print #1 DATA&

Rem 開いたファイルを閉じます
Close #1
```

・使用可能なモジュール
ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「Close (P.8-45)」
「GetSystemData (P.8-133)」
「Input$ (P.8-150)」
「Line Input# (P.8-165)」
「Open For Input As# (P.8-193)」
「Eof (P.8-104)」
「Input# (P.8-149)」
「Isfile (P.8-154)」
「Open For Append As# (P.8-192)」
「Print# (P.8-203)」

OpenTextData

メッセージファイルを開きます。

・書式

OpenTextData <ident> As #<textDataNo>

・パラメータ

パラメータ名	データ型	説明
<Ident>	文字列型	テキストデータとして開くメッセージファイルの識別名
<textDataNo>	整数型	テキストデータとして開くメッセージファイルに割り当てるテキストデータ番号（0～15）

・戻り値

なし

・解説

- メッセージファイルは、画面に表示したいメッセージを定義するファイルです。言語ごとに 1 ファイル作成されています。
- メッセージファイルのファイル名は、<メッセージファイル識別名>_<言語識別名>.msg で構成されています。
  <メッセージファイル識別名> は、OpenTextData 関数で<ident> として指定するメッセージファイルの識別名です。
  <言語識別名> は、言語を表す文字列です。
  簡体中国語：chs
  繁体中国語：cht
  ドイツ語：deu
  英語：eng
  スペイン語：esp
  フランス語：fra
  イタリア語：ita
  日本語：jpn
  韓国語：kor
- メッセージファイルの書式は、1 行ごとに<textIdent> と指定する識別名と、それに対応するテキストデータを記述されています。
  <識別名 1> = <テキストデータ 1>
  <識別名 2> = <テキストデータ 2>
  <識別名 3> = <テキストデータ 3>
  ：
  <識別名 n> = <テキストデータ n>

<ident> パラメータで指定したファイルを、<textDataNo> パラメータで指定したテキストデータ番号を割り当てて開きます。
このマクロ関数を実行して開いたメッセージファイルは、必ず CloseTextData 関数を実行して同じ処理の中で閉じてください。CloseTextData 関数を実行しない場合やこのマクロ関数を実行したサブルーチンとは別のサブルーチンや実行タイミングで実行した場合は、メッセージファイルを正しく閉じられず、その後の処理で正しくこのマクロ関数を実行できない可能性があります。
<textDataNo> パラメータとして、0～15 の範囲を超える値を指定した場合は、Illegal function call のエラーが発生します。
<textDataNo> パラメータとして、すでに開いているテキストデータ番号を指定した場合は、Illegal function call のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。

文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
ユニットマクロ処理ユニットの*MEASUREDISPT サブルーチンの中で、処理ユニット番号 5 のサーチ処理ユニットで計測した相関値を、処理ユニットのメッセージファイルから取得した文字列とともにテキスト表示ウィンドウに表示します。相関値は外部参照データ番号 5 です。

```
*MEASUREDISPT

    Rem 計測結果を取得します
    GetUnitData 5, 5, CR#

    Rem メッセージファイルを開きます
    OpenTextData "Search" As #1

    Rem テキストを取得します
    TEXT$ = GetText$(#1, "Correlation")

    Rem テキスト表示ウィンドウにメッセージファイルから取得した文字列を改行なしで描画します
    DrawText TEXT$, UnitJudge(5), 0

    Rem テキスト表示ウィンドウに計測結果を描画します
    DrawText Str2$(CR#, 4, 4, 0, 0), UnitJudge(5), 1

    Rem メッセージファイルを閉じます
    CloseTextData

Return
```

結果は以下のとおりになります。

```
相関値：90.0000
```

・使用可能なモジュール
ユニット演算マクロ ／ ユニットマクロ

・対応バージョン
バージョン 5.00 以降

・関連項目

「CloseTextData (P.8-46)」
「DrawText (P.8-97)」
「GetText$ (P.8-134)」
「GetUnitData (P.8-138)」
「UnitJudge (P.8-317)」

Option Explicit

Dim 変数の書式で定義した変数の未定義や重複のチェックを実行します。

・書式

Option Explicit

・パラメータ

なし

・戻り値

なし

・解説

Option Explicit コマンドを実行すると、Dim 変数の書式で定義した変数のみを使用可能とし、定義した変数の重複をチェックします。

・使用上の注意

- Option Explicit コマンドは、コマンドを実行した以降の行で、チェック機能が動作します。
- コマンド実行の行よりも前で使用している変数は、変数の未定義や重複のチェック機能の範囲外となります。そのため、コマンド実行の前に使用している変数は、未定義や重複エラーにはなりません。
- Option Explicit コマンドは、電源投入時やマクロのロード時には無効状態です。マクロが実行され、Option Explicit が実行された後、有効になります。その後、次のマクロがロードあるいは電源 OFF まで有効になります。

・使用例

マクロ変数のチェックを行います。

```
Option Explicit
Rem Dim 変数による変数の定義
Dim A&
Dim B&
Dim C&
A& = 0
B& = 1
C& = A& + B&
Print Str$(C&)
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 5.40 以降

・関連項目

「Dim (P.8-67)」
「ReDim (P.8-214)」
「VarList (P.8-322)」

OR

2 式の論理和を求めます。

・書式

<expression1> OR <expression2>

・パラメータ

パラメータ名	データ型	説明
<expression1>	整数型	論理和を求める数式
<expression2>	整数型	

・戻り値

整数型の論理和の値を返します。

・解説

<expression1> パラメータで指定した数式と、<expression2> パラメータで指定した数式のビットごとの論理和を求めます。
<expression1> パラメータと<expression2> パラメータの値が実数型の場合、小数部を四捨五入した値として扱われます。
If 文の中で Or 条件としても使用できます。論理式の詳細については「4-1-5 演算子 (P.4-12)」を参照してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

変数 EXP1& に代入した 1 と変数 EXP2& に代入した 4 の値の論理和を求めます。

```
EXP1& = 1
EXP2& = 4

EXPALL& = EXP1& OR EXP2&
```

結果は以下のとおりになります。

```
EXPALL& = 5
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「AND (P.8-25)」
「UnitData (P.8-311)」
「XOR (P.8-331)」
「GetUnitData (P.8-138)」
「NOT (P.8-191)」

Piece$

文字列から区切り文字で区切られた部分文字列を取り出します。

・書式

Piece$(<string>, <delimiter>, <start>, <end>)

・パラメータ

パラメータ名	データ型	説明
<string>	文字列型	取り出し元の文字列
<delimiter>	文字列型	区切り文字
<start>	整数型	取り出しを開始する文字列のインデックス番号（1～部分文字列の数）
<end>	整数型	取り出しを終了する文字列のインデックス番号（1～部分文字列の数）

・戻り値

文字列型の取り出した値を返します。

・解説

<string> パラメータで指定した文字列を、<delimiter> パラメータで指定した区切り文字で部分文字列に区切ってインデックスを付与し、<start> パラメータで指定したインデックス番号から、<end> パラメータで指定したインデックス番号までの部分文字列を取り出します。
<string> パラメータで指定した文字列を<delimiter> パラメータで指定した文字で区切れなかった場合は、部分文字列としてすべての文字列を取り出します。
<string> パラメータとして、255 文字を超える文字列を指定した場合は、256 文字め以降を切り捨てた 255 文字の文字列として扱われます。
<delimiter> パラメータとして、1 文字を超える文字列を指定した場合は、部分文字列の取り出しが正しく行えません。
<start> パラメータには、取り出しを開始する部分文字列のインデックス番号を指定してください。
インデックス番号は、<string> パラメータで指定した文字列を<delimiter> パラメータで指定した区切り文字で先頭から順に区切った部分文字列群の最初の部分文字列を 1 として、昇順に割り振られます。
<start> パラメータとして、区切られた部分文字列の数より大きな値を指定した場合は、Illegal function call のエラーが発生します。
<start> パラメータで指定したインデックス番号が<end> パラメータで指定したインデックス番号よりも大きい場合は、Illegal function call のエラーが発生します。
<start> パラメータまたは<end> パラメータとして、0 以下の値を指定した場合は、Illegal function call のエラーが発生します。
<end> パラメータとして、区切られた部分文字列の数より大きな値を指定した場合は、<start> パラメータで指定したインデックス番号以降の部分文字列すべてを取り出します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

区切り文字としてセミコロン(;) で区切られた部分文字列を取得します。

```
INPUTSTR$ = "PIECE1;PIECE2;PIECE3;PIECE4"
DELIMITER$ = ";"

Rem 最初の部分文字列を取り出します
OUTPUTSTR1$ = Piece$(INPUTSTR$, DELIMITER$, 1, 1)
Rem 3 番目の部分文字列から 4 番目の部分文字列までを取り出します
OUTPUTSTR2$ = Piece$(INPUTSTR$, DELIMITER$, 3, 4)
```

結果は以下のとおりになります。

```
OUTPUTSTR1$ = "PIECE1"
OUTPUTSTR2$ = "PIECE3;PIECE4"
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「Asc (P.8-31)」
「Chr$ (P.8-41)」
「Hex$ (P.8-142)」
「LCase$ (P.8-162)」
「Left$ (P.8-163)」
「Len (P.8-164)」
「Mid$ (P.8-186)」
「Right$ (P.8-223)」
「Str$ (P.8-287)」
「Str2$ (P.8-288)」
「UCase$ (P.8-309)」
「Val (P.8-321)」

Print

システム状態監視コンソールウィンドウにデータを出力します。

・書式

Print <expression>[;¦, <expression>...][;¦,]

・パラメータ

パラメータ名	データ型	説明
<expression>	整数型 実数型 文字列型 配列型	出力する数式や文字列

・戻り値

なし

・解説

<expression> パラメータで指定した数式や文字列を、システム状態監視コンソールウィンドウに出力します。(詳細は、「3-1-2 システム状態監視コンソールウィンドウの見かた (P.3-6)」を参照してください。)

パラメータをカンマ(,) で区切った場合は、<expression> パラメータで指定された数式や文字列ごとにタブ区切りで出力します。

パラメータをセミコロン(;) で区切った場合は、<expression> パラメータで指定された数式や文字列を直前に出力した数式や文字列の後に続けて出力します。

パラメータの末尾にセミコロン(;) とカンマ(,) のいずれも指定しない場合は、すべてのパラメータを出力したあとシステム状態監視コンソールウィンドウの表示を改行します。

マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- システム状態監視コンソールウィンドウにデータを出力すると、システム状態監視コンソールウィンドウがセンサコントローラのメイン画面の前面に表示されます。メイン画面を前面に表示する場合は、システム状態監視コンソールウィンドウ右上の[_]をクリックするか、センサコントローラに USB キーボードを接続して、[Alt] + [Tab] で表示する画面を切り替えてください。

・使用例

処理ユニット番号 2 のサーチ処理ユニットの「相関値」「計測座標 X」「計測座標 Y」を取得してシステム状態監視コンソールウィンドウに出力します。相関値は外部参照データ番号 5、計測座標 X は外部参照データ番号 6、計測座標 Y は外部参照データ番号 7 です。

```
Rem 計測結果を取得します
GetUnitData 2, 5, CR#
GetUnitData 2, 6, X#
GetUnitData 2, 7, Y#

Rem 取得した計測結果をシステム状態監視コンソールウィンドウに出力します
Print CR#; ","; X#; ","; Y#
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「Cls (P.8-48)」
「Cont (P.8-49)」
「Debug (P.8-63)」
「DebugPrint (P.8-64)」
「GetUnitData (P.8-138)」
「List (P.8-167)」
「SetStop (P.8-270)」
「SetVar (P.8-281)」
「Stop (P.8-286)」
「SubList (P.8-292)」
「VarList (P.8-322)」

Print#

ファイルにデータを出力します。

・書式

Print #<fileNo>[, <expression>[;¦, <expression>...][;¦,]

・パラメータ

パラメータ名	データ型	説明
<fileNo>	整数型	出力先ファイルのファイル番号（0～15）
<expression>	整数型 実数型 文字列型 配列型	出力する数式や文字列

・戻り値

なし

・解説

<fileNo> パラメータで指定したファイル番号のファイルに、<expression> パラメータで指定した数式や文字列を出力します。
パラメータをカンマ(,) で区切った場合は、<expression> パラメータで指定された数式や文字列ごとにタブ区切りで出力します。
パラメータをセミコロン(;) で区切った場合は、<expression> パラメータで指定された数式や文字列を直前に出力した数式や文字列の後に続けて出力します。
パラメータの末尾にセミコロン(;) とカンマ(,) のいずれも指定しない場合は、すべてのパラメータを出力したあと改行します。
<fileNo> パラメータとして、開かれていないファイル番号を指定した場合は、Illegal function call のエラーが発生します。
<fileNo> パラメータとして、Open For Append As# 関数および Open For Output As# 関数以外のマクロ関数で開かれたファイルのファイル番号を指定した場合は、Illegal function call のエラーが発生します。
<fileNo> パラメータとして、0～15 の範囲を超える値を指定した場合は、Illegal function call のエラーが発生します。
<expression> パラメータとして、127 文字を超える文字列を指定した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

ファイル「E:¥input.txt」に文字列を出力します。

```
STRING$ = "Sample"

Rem ファイルを開きます
Open "E:¥input.txt" For Append As #1

Print #1, STRING$

Rem ファイルを閉じます
Close #1
```

・使用可能なモジュール
ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「Close (P.8-45)」
「Open For Append As# (P.8-192)」
「Open For Output As# (P.8-194)」

PutAll

すべての出力端子の出力状態を設定します。

・書式
PutAll <ioIdent>, <state>

・パラメータ

パラメータ名	データ型	説明
<ioIdent>	文字列型	使用する通信モジュールの識別名 （常に"Parallello"） （参照：「9-4 IO モジュール一覧 (P.9-25)」）
<state>	整数型	端子の出力状態

・戻り値
なし

・解説
<ioIdent> パラメータで指定した通信モジュールのすべての出力端子の出力状態を、<state> パラメータで指定した出力状態の値に設定します。
<ioIdent> パラメータには、通常は"Parallello" を指定してください。
出力状態は、出力端子ごとの OFF(0) / ON(1) を 2 進数表記の各桁で表現した整数の値です。
パラレル I/O の場合は、DO0 から DO15 を 1 桁めから 16 桁めで表現した整数の値を返します。
（例）DO0 から DO5 が ON、DO6～DO15 が OFF の場合
2 進数表記：0000 0000 0011 1111
設定する出力状態の値：63
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- 動作モードが複数ラインランダムトリガの場合でライン数が 3 以上の場合は、DO 信号の出力状態を設定することはできません。ライン数が 3 ライン以上の場合にこのマクロ関数を使用しようとした場合でも、Illegal function call のエラーは発生しません。

・使用例

通信コマンドマクロで、パラレル I/O の DO 出力状態をすべて ON に設定します。

```
IOMODULE$ = "Parallel1o"

Rem 出力状態を設定します
PutAll IOMODULE$, 65535
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「BusyOut (P.8-34)」
「GetPort (P.8-129)」
「PutPort (P.8-205)」
「GetAll (P.8-120)」
「JudgeOut (P.8-159)」
「RunOut (P.8-225)」

PutPort

指定した出力端子の出力状態を設定します。

・書式

PutPort <ioIdent>, <portNo>, <state>

・パラメータ

パラメータ名	データ型	説明
<ioIdent>	文字列型	使用する通信モジュールの識別名 （常に"Parallello"） （参照：「9-4 IO モジュール一覧 (P.9-25)」）
<portNo>	整数型	出力状態を設定する出力端子の端子番号 パラレル I/O の場合 • FH DO0～DO15：0～15 GATE N：100+N×8（N：ライン番号（0～7）） BUSY N：101+N×8 OR N：102+N×8 ERROR N：103+N×8 RUN N：104+N×8 READY N：105+N×8 ACK：200
<state>	整数型	出力端子の出力状態 0：出力 OFF 状態 1：出力 ON 状態

・戻り値

なし

・解説

<ioIdent> パラメータで指定した通信モジュールの、<portNo> パラメータで指定した端子番号の出力端子の状態を、<state> パラメータで指定した出力状態の値に設定します。
<ioIdent> パラメータには、通常は"Parallello" を指定してください。

パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
<ioIdent> パラメータと<portNo> パラメータとして、存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
<state> パラメータとして、存在しない出力状態の値を指定しても、エラーは発生しません。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- 他のパラレルの出力制御と同時に実行しないでください。

・使用例

通信コマンドマクロで、パラレル I/O の DO3 の出力状態を ON に設定します。

```
IOMODULE$ = "Parallel1o"

Rem 出力状態を設定します
PutPort IOMODULE$, 3, 1
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「BusyOut (P.8-34)」
「GetAll (P.8-120)」
「GetPort (P.8-129)」
「JudgeOut (P.8-159)」
「PutAll (P.8-204)」
「RunOut (P.8-225)」

RaiseErrorProcEvent

UI 画面にエラー処理イベントを通知します。

・書式

RaiseErrorProcEvent <errorKind>, <parameter>[, <wait>]

・パラメータ

パラメータ名	データ型	説明
<errorKind>	整数型	通知するエラー処理イベントのエラー種別番号 0：システム異常 1：システム異常（ファン／電圧異常） 3：バッテリ異常 4：UWF オーバレイ枯渇 10：カメラ接続異常 11：接続カメラに変更あり 12：カメラ過電流検出 13：照明機器接続構成異常 20：ロギングディスク書込異常 30：パラレル出力タイムアウト 31：PLC リンク通信エラー 32：パラレル I/O 過電流検出 40：データロード異常 41：データ転送異常 42：起動時シーングループ番号不正 43：起動時シーン番号不正
<parameter>	整数型	通知するエラー処理のパラメータ 現時点では未使用。そのため常に 0 を指定。
<wait>	整数型	マクロ関数の復帰タイミング 0：計測終了を待たずに以降のプログラム行を実行します。 1：計測終了を待ってから以降のプログラム行を実行します。 2：計測終了と計測結果表示の終了を待ってから以降のプログラム行を実行します。 3：計測終了、計測結果表示の終了、UI 表示更新の終了を待ってから以降のプログラム行を実行します。

・戻り値

なし

・解説

<errorKind> で指定したエラー種別番号を、<parameter> で指定したイベントのパラメータとともに、UI 画面（UI プロセスである PanDA）に対して、エラー処理イベントを通知します。エラー処理イベントを受信した UI 画面でどのような処理が実行されるかは、UI 画面のプロセスの仕様に依存します。

<parameter> は、エラー処理のパラメータですが、現時点で未使用のため、常に 0 を指定します。

<wait> パラメータは省略可能です。

パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。

パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定しても、Illegal function call のエラーは発生しません。

整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。

マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- *MCRINIT に記述しないでください。

・使用例

通信コマンドマクロの使用例です。
ロギングエラー発生時は、ERROR 信号を ON せずにエラー処理のイベントを通知し、他のエラーの場合は通常のエラー処理を行います。

```
*ERRORPROC

    If ERRORKIND& <>20 Then
       Rem ロギングエラー以外のエラーの場合、ERROR 信号の状態を ON にする。
       ERROROUT "",ERRORKIND&
    Endif

    Rem UI 画面にエラー処理のイベントを通知する。
    RaiseErrorProcEvent ERRORKIND&, 0, 0

Return
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン

バージョン 5.40 以降

・関連項目

「ErrorOut (P.8-108)」
「If Then Else (P.8-143)」
「MessageBox (P.8-185)」

RaiseOptionEvent

UI 画面にオプションイベントを通知します。

・書式

RaiseOptionEvent <eventNo>, <parameter>

・パラメータ

パラメータ名	データ型	説明
<eventNo>	整数型	通知するイベントのイベント番号 255：センサコントローラとリモートオペレーションツール両方のメイン画面のレイアウト切替 511：センサコントローラのメイン画面のレイアウト切替 767：リモートオペレーションツールのメイン画面のレイアウト切替
<parameter>	整数型	通知するイベントのパラメータ イベントがレイアウト切替の場合 0～127：切替先のレイアウト番号

・戻り値

なし

・解説

<eventNo> パラメータで指定したイベントを、<parameter> パラメータで指定したパラメータとともに UI 画面に対して通知します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定しても、Illegal function call のエラーは発生しません。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。
Application Producer（形 FH-AP1、別売）を使用することで、UI 画面上で新しくイベントを作成することができます。

・使用上の注意

このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）

・使用例

通信コマンドマクロで、BUSY ON の状態の時に、センサコントローラのメイン画面のレイアウトを切り替えます。

```
Rem レイアウト 1 に切り替えます
RaiseOptionEvent 511, 1
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

なし

ReadPlcMemory

PLC のメモリ領域から値を読み込みます。

・書式

ReadPlcMemory <ioIdent>, <area>, <channelOffset>, <channelCount>, <readData()>

・パラメータ

パラメータ名	データ型	説明
<ioIdent>	文字列型	使用する通信モジュールの識別名 （参照：「9-4 IO モジュール一覧 (P.9-25)」）
<area>	整数型	読み込み対象のデータ出力エリアのエリア種別番号
<channelOffset>	整数型	データ出力エリア先頭から読み込み開始するアドレスまでのオフセット
<channelCount>	整数型	読み込むデータのサイズ（チャネル単位）
<readData()>	整数型配列	読み込んだデータ

・戻り値

なし

・解説

<ioIdent> パラメータで指定した通信モジュールを使用して、<area> パラメータで指定した PLC のエリア種別の、<channelOffset> パラメータで指定した値だけオフセットしたアドレスから、<channelCount> パラメータで指定したサイズ分のデータを読み込みます。
このマクロ関数で PLC のメモリ領域から値を読み出した後、GetPlcData 関数を実行して読み込んだ値を取り出してください。
<readData()>パラメータには、読み込んだデータを格納する 1 次元の整数型配列変数を、要素番号を指定せず( )だけ付加して指定してください。
<area>パラメータには、PLC リンクの場合は、システム設定の PLC リンク設定で設定したレジスタの ID を指定してください。Fieldbus の場合は、エリア種別を指定できないため 0 を指定してください。
<channelCount> パラメータには、チャネル単位のサイズを指定してください。整数型は 1 データが 2 チャネル分（4 バイト）のサイズのため、整数の値を 1 個読み込む場合は、<readData()>パラメータに要素数 1 の配列を用意して、<channelCount>パラメータには 2 を指定してください。
<channelCount>パラメータに、<readData()>パラメータで指定した配列のサイズより大きなサイズを指定した場合は、Subscript out of range のエラーが発生します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数を使用してデータを読み込んだ場合は、必ず GetPlcData 関数を使用して読み込んだデータから値を取り出してください。GetPlcData 関数を使用せずに<readData()> パラメータから値を取得する場合は、正しい値を取得できない可能性があります。

・使用例

通信コマンドマクロで、PLC リンクで接続された PLC から複数のデータを読み込みます。

```
IOMODULE$ = "UdpPlcLink"

Rem データ出力エリアの設定を取得します
GetSystemData IOMODULE$, "outputArea", AREA&
GetSystemData IOMODULE$, "outputMemoryAddress", ADDRESS&

Rem 読み込むデータを格納する整数型配列変数を作成します
Dim DATA&(1)

Rem データ出力エリアからデータ（4CH 分）を読み込みます
ReadPlcMemory IOMODULE$, AREA&, ADDRESS&, 4, DATA&()

Rem 読み込んだデータから値を取得します
GetPlcData IOMODULE$, DATA&(), 0, 4, VALUE0&
GetPlcData IOMODULE$, DATA&(), 4, 4, VALUE1&
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 4.20 以降

・関連項目

「GetPlcData (P.8-126)」
「SetPlcData (P.8-258)」
「WritePlcMemory (P.8-329)」

ReceiveData

データを受信します。

・書式

ReceiveData <ioIdent>, <inputData()>, <inputMaxSize>, <inputSize>[, <parameter()>, <parameterSize>]

・パラメータ

パラメータ名	データ型	説明
<ioIdent>	文字列型	使用する通信モジュールの識別名 （参照：「9-4 IO モジュール一覧 (P.9-25)」）
<inutData()>	整数型配列	受信したデータ
<inputMaxSize>	整数型	受信するデータの最大サイズ
<inputSize>	整数型	受信したデータのサイズ
<parameter()>	整数型配列	オプションで指定するパラメータデータ
<parameterSize>	整数型	オプションで指定するパラメータデータのサイズ

・戻り値

なし

・解説

<ioIdent> パラメータで指定した通信モジュールを使用して、<imputMaxSize> パラメータで指定したサイズまでデータを受信します。
<inputData()> パラメータには、受信したデータを格納する 1 次元の整数型配列変数を、要素番号を指定せず() だけ付加して指定してください。<inputSize> パラメータには、受信したデータのサイズを格納する整数型の変数を指定してください。
<parameter()> パラメータと<parameterSize> パラメータに設定できる値は、<ioIdent> パラメータで指定した通信モジュールによって異なります。<parameter()> パラメータと<parameterSize> パラメータは省略可能です。
このマクロ関数では、マクロ関数を実行した時点で外部装置からセンサコントローラに到着しているデータを受信して、到着済のデータが存在しない場合は<inputSize> パラメータに 0 が格納されます。受信したいデータを 1 回で受信できない場合は、すべてのデータを受信するまでこのマクロ関数を繰り返し実行してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
ハンドシェイクありの設定で通信がタイムアウトした場合などデータ受信に失敗した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、対象の通信モジュールの実行状態が停止状態で実行してください。（詳細は、「5-1-5 処理の排他制御 (P.5-18)」を参照してください。）

・使用例

ユニットマクロの計測処理において外部機器からデータを受信します。

```
Rem データを受信するために 12 バイト受信できるバッファを用意します
Dim BUFFER&(11)
IOMODULE$ = "TcpNormal"

Rem データを受信するために通信モジュールの実行状態を停止状態に設定します
SetPollingState IOMODULE$, False

Rem 受信データサイズを初期化します
SIZE& = 0

Rem データを受信するまで受信処理を繰り返します
Try
   Do
      Rem データの受信を試みます
      ReceiveData IOMODULE$, BUFFER&(), 12, SIZE&

      Rem データを受信したらデータサイズをシステム状態監視コンソールウィンドウに表示します
      If(SIZE& > 0) Then
         Print "Received data size = " + Str$(SIZE&)
      Endif
   Loop While SIZE& = 0

   Rem データを受信したので通信モジュールの実行状態を動作状態に設定します
   SetPollingState IOMODULE$, True

Catch
   Rem 停止状態にしている通信モジュールの実行状態を動作状態に戻します
   If GetPollingState(IOMODULE$) = False Then
      SetPollingState IOMODULE$, True
   Endif

End Try
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「GetPollingState (P.8-127)」
「SendData (P.8-246)」
「SendString (P.8-248)」
「SetPollingState (P.8-260)」

ReDim

配列変数を再定義します。

・書式

ReDim <arrayName>(<maxCount>[, <maxCount>[, <maxCount>[, <maxCount>]]])

・パラメータ

パラメータ名	データ型	説明
<arrayName>	-	使用する配列の変数名
<maxCount>	整数型	添字の最大値

・戻り値

なし

・解説

<arrayName> パラメータで指定した変数名の、<maxCount> パラメータで指定した添字の最大値を持つ 1 次元から 4 次元までの配列変数を再定義します。
<arrayName> パラメータの末尾には、配列変数のデータ型を意味する型識別子を指定してください。(詳細は、「4-1-3 変数 (P.4-6)」を参照してください。)
このマクロ関数で定義した配列変数は、Erase 関数を実行して解放してください。
次元が異なる配列変数は、同じ変数名であれば同じ変数として扱われます。
配列変数と変数は、同じ変数名であっても異なる変数として扱われます。
配列変数の要素数が未定義の場合、Syntax error が発生します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- Option Explicit コマンドの使用によって振る舞いが変わります。
  Option Explicit コマンド使用時、事前に Dim 関数で定義されていない変数を再定義した場合は、マクロロード時に未定義のエラーが発生します。
  Option Explicit コマンド不使用時、事前に Dim 関数の定義は必要ありません。Dim 関数と同様に配列変数を定義します。

・使用例

配列を再定義します。

```
Dim AA&()
ReDim AA&(10)
Dim BB&(3)
Redim BB&(10)
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 5.40 以降

・関連項目

「Dim (P.8-67)」
「Erase (P.8-105)」
「Option Explicit (P.8-198)」

RefreshImageWindow

画像表示ウィンドウの表示を更新します。

・書式
RefreshImageWindow

・パラメータ
なし

・戻り値
なし

・解説
画像表示ウィンドウの表示を更新します。
画像表示ウィンドウ上に表示しているグラフィックを再描画する場合や画像を再描画する場合、ImageUpdate 関数を実行した後などに、このマクロ関数を実行して表示を更新してください。

・使用上の注意
なし

・使用例
通信コマンドマクロで、画像モードが「カメラ画像フリーズ」で表示された画像を最新の画像に更新します。

```
Rem フリーズ表示画像を最新の画像に更新します
ImageUpdate

Rem 更新した画像を画像表示ウィンドウの表示に反映します
RefreshImageWindow
```

・使用可能なモジュール
シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「ImageUpdate (P.8-147)」
「RefreshJudgeWindow (P.8-216)」
「RefreshTextWindow (P.8-217)」
「RefreshTimeWindow (P.8-218)」
「SetImageWindow (P.8-254)」

RefreshJudgeWindow

判定表示ウィンドウの表示を更新します。

・書式
RefreshJudgeWindow

・パラメータ
なし

・戻り値
なし

・解説
判定表示ウィンドウの表示を更新します。

・使用上の注意
なし

・使用例
通信コマンドマクロで、判定表示ウィンドウの表示を更新します。

```
RefreshJudgeWindow
```

・使用可能なモジュール
シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「RefreshImageWindow (P.8-215)」
「RefreshTextWindow (P.8-217)」
「RefreshTimeWindow (P.8-218)」

RefreshTextWindow

テキスト表示ウィンドウの表示を更新します。

・書式
RefreshTextWindow

・パラメータ
なし

・戻り値
なし

・解説
テキスト表示ウィンドウの表示を更新します。
テキスト表示ウィンドウ上に表示している詳細結果を再描画する場合に、このマクロ関数を実行して表示を更新してください。

・使用上の注意
なし

・使用例
通信コマンドマクロで、現在のシーンの処理ユニット番号 5 の演算処理ユニットの演算式 0 の計測結果に値を設定した後に、テキスト表示ウィンドウの表示を更新します。「演算式 0 の演算結果」は外部参照データ番号 5 です。

```
Rem 値を計測結果として設定します
SetUnitData 5, 5, 100

Rem テキスト表示ウィンドウの表示に反映します
RefreshTextWindow
```

・使用可能なモジュール
シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「RefreshImageWindow (P.8-215)」
「RefreshJudgeWindow (P.8-216)」
「RefreshTimeWindow (P.8-218)」
「SetTextWindow (P.8-274)」

RefreshTimeWindow

情報表示ウィンドウの表示を更新します。

・書式
RefreshTimeWindow

・パラメータ
なし

・戻り値
なし

・解説
情報表示ウィンドウの表示を更新します。

・使用上の注意
なし

・使用例
通信コマンドマクロで、情報表示ウィンドウの表示を更新します。

```
RefreshTimeWindow
```

・使用可能なモジュール
シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「RefreshImageWindow (P.8-215)」
「RefreshJudgeWindow (P.8-216)」
「RefreshTextWindow (P.8-217)」
「SetTextWindow (P.8-274)」

Rem

プログラムに注釈を入れます。

・書式

Rem

・パラメータ

なし

・戻り値

なし

・解説

プログラムに注釈コメントや説明を入れます。注釈コメントを入れることでプログラムの可読性が向上します。
注釈コメントの詳細は、「注釈コメント (P.4-4)」をご覧ください。

・使用上の注意

- 注釈コメントは注釈コメント以外のステートメントと 1 行に混在させないでください。注釈コメントとそれ以外のステートメントが 1 行に混在した場合は、注釈コメントが正しく認識されずプログラムが誤動作を起こす可能性があります。

・使用例

処理内容を説明する注釈コメントをプログラムに追加します。

```
Rem 直近の処理ユニットの判定結果をシステム状態監視コンソールウィンドウに表示します
Print UnitJudge(UnitNo - 1)
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「Print (P.8-202)」
「UnitJudge (P.8-317)」
「UnitNo (P.8-318)」

Remeasure

再計測処理を実行します。

・書式

Remeasure <preImageNo>[, <wait>]
Remeasure <fileName>[, <wait>]

・パラメータ

パラメータ名	データ型	説明
<preImageNo>	整数型	再計測対象の画像の画像ロギング番号（0～すでにロギングしている本体ロギング画像の枚数）
<fileName>	文字列型	再計測対象の画像の画像ファイル名
<wait>	整数型	マクロ関数の復帰タイミング 0：計測終了を待たずに以降のプログラム行を実行します。 1：計測終了を待ってから以降のプログラム行を実行します。 2：計測終了と計測結果表示の終了を待ってから以降のプログラム行を実行します。 3：計測終了、計測結果表示の終了、UI 表示更新の終了を待ってから以降のプログラム行を実行します。

・戻り値

なし

・解説

<preImageNo> パラメータで指定した画像ロギング番号の画像または<fileName> パラメータで指定した画像ファイル名の画像に対して、<wait> パラメータで指定した復帰タイミングで再計測処理を実行します。
<preImageNo> パラメータには、本体ロギング画像としてすでにロギングしているロギング画像の番号を指定してください。（詳細は、『画像処理システム FH/FHV シリーズ ユーザーズマニュアル (SDNB-719)』の「ロギング条件を設定する［ロギング設定］」を参照してください。）
<preImageNo> パラメータとして、0 を指定した場合は、最新の本体ロギング画像に対して再計測を実行します。
<fileName> パラメータには、再計測対象の画像のファイル名を絶対パスで指定してください。
<wait> パラメータを省略した場合は、<wait> パラメータに 0 を設定したときと同じ動作になります。
<wait> パラメータとして 0 を指定した場合は、このマクロ関数の実行直後に実行した計測処理が終了していない可能性があるため、計測結果を正しく取得できません。計測結果を取得したい場合は、<wait> パラメータに 1、2、3 のいずれかを指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定しても、Illegal function call のエラーは発生しません。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、BUSY 信号などの計測中信号が OFF した計測許可状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）

・使用例

通信コマンドマクロで、最新のロギング画像を使用して再計測処理を実行した後に、処理ユニット番号 2 のサーチ処理ユニットの計測 X 座標と計測 Y 座標を取得します。計測 X 座標は外部参照データ番号 6、計測 Y 座標は外部参照データ番号 7 です。

```
Rem 再計測を実行して計測終了まで待ちます
Remeasure 0, 1

Rem 計測結果を取得します
GetUnitData 2, 6, POSX#
GetUnitData 2, 7, POSY#
```

・使用可能なモジュール
通信コマンドマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「Measure (P.8-176)」　「MeasureStart (P.8-182)」
「MeasureStop (P.8-183)」

RenumUnitNo

フロー編集後の処理ユニット番号を取得します。

・書式
RenumUnitNo(<oldUnitNo>)

・パラメータ

パラメータ名	データ型	説明
<oldUnitNo>	整数型	変更前の処理ユニット番号（0～（現在のシーンの処理ユニット数-1)）

・戻り値
整数型の計測フロー更新後の処理ユニット番号の値を返します。

・解説
<oldUnitNo> パラメータで指定した処理ユニットの、フロー編集後に移動した先の処理ユニット番号を取得します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定しても、Illegal function call のエラーは発生しません。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。
フロー編集のときの処理ユニット番号の変化に対処する方法として、このマクロ関数を使う以外に参照変数を使う方法があります。(詳細については、「4-1-3 変数 (P.4-6)」を参照してください。)

・使用上の注意
- このマクロ関数は、*RENUMPROC サブルーチンの中でしか使用できません。それ以外のサブルーチンで使用した場合は、Illegal function call のエラーが発生します。

・使用例

ユニットマクロの*RENUMPROC サブルーチンの中で、処理ユニット番号 5 の処理ユニットがフロー編集後に移動した先の処理ユニット番号を取得します。

```
*RENUMPROC

    LATESTUNITNO& = 5

    Rem フロー編集後の処理ユニット番号を取得します
    LATESTUNITNO& = RenumUnitNo(LATESTUNITNO&)

Return
```

・使用可能なモジュール

ユニット演算マクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「DeleteUnit (P.8-66)」
「InsertUnit (P.8-152)」
「UnitNo (P.8-318)」
「Ut (P.8-320)」

RGB

カラー値を表す数値を取得します。

・書式

RGB(<red>, <green>, <blue>)

・パラメータ

パラメータ名	データ型	説明
<red>	整数型	カラー値を求める色の赤の成分（0～255）
<green>	整数型	カラー値を求める色の緑の成分（0～255）
<blue>	整数型	カラー値を求める色の青の成分（0～255）

・戻り値

整数型のカラー値の値を返します。
カラー値の下位 1 バイトには赤の成分、中位の 1 バイトには緑の成分、上位側の 1 バイトには青の成分の値が格納されます。

・解説

<red> パラメータで指定された赤の成分と、<green> パラメータで指定された緑の成分と、<blue> パラメータで指定された青の成分を持つ色のカラー値を取得します。描画スタイルを設定するマクロ関数や一部の描画マクロ関数では、パラメータにカラー値を指定することで描画色を指定することができます。
<red> パラメータ、<green> パラメータ、<blue> パラメータに設定可能範囲を超えた値を指定した場合は、予期しない値を返します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定しても、Illegal function call のエラーは発生しません。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
テキスト表示ウィンドウに緑色の文字列を表示します。

```
Rem 緑色のカラー値を取得します
COLOR& = RGB(0, 255, 0)

Rem テキスト表示ウィンドウに文字列を描画します
DrawText "Processing OK", COLOR&, 1
```

・使用可能なモジュール
ユニット演算マクロ ／ ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「DrawFillImage (P.8-86)」
「DrawText (P.8-97)」
「SetDrawStyle (P.8-250)」
「SetTextStyle (P.8-272)」

Right$

文字列の右側から指定した長さの文字列を取り出します。

・書式
Right$(<string>, <length>)

・パラメータ

パラメータ名	データ型	説明
<string>	文字列型	取り出し元の文字列
<length>	整数型	取り出す文字列の長さ（1～取り出し元の文字列の長さ）

・戻り値
文字列型の取り出した値を返します。

・解説
<string> パラメータで指定した文字列の先頭である右側から、<length> パラメータで指定した長さの文字列を取り出します。
<length> パラメータには、取り出す文字数を文字のバイト数で指定してください。バイト数は、半角英数字や半角記号などの 1 バイト文字では 1 文字あたり 1 バイト、2 バイト文字では 1 文字あたり 2 バイトの値です。
<length> パラメータで指定した長さが<string> パラメータで指定した文字列より長い場合は、<string> パラメータで指定した文字列すべてを取り出します。
<length> パラメータとして、0 以下の値を指定した場合は、Illegal function call のエラーが発生します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定しても、エラーは発生しません。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
文字列の末尾から 6 バイト分の文字列を取り出します。半角英字は 1 文字あたり 1 バイトなので、6 文字分の文字列を取り出すことができます。

```
CHARA$ = "Measurement Result"

Rem 文字列の末尾から半角英字 6 文字分の文字列を取り出します
TITLE$ = Right$(CHARA$, 6)
```

結果は以下のとおりになります。

```
TITLE$ = "Result"
```

・使用可能なモジュール
ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「Asc (P.8-31)」
「Chr$ (P.8-41)」
「Hex$ (P.8-142)」
「LCase$ (P.8-162)」
「Left$ (P.8-163)」
「Len (P.8-164)」
「Mid$ (P.8-186)」
「Piece$ (P.8-200)」
「Str$ (P.8-287)」
「Str2$ (P.8-288)」
「UCase$ (P.8-309)」
「Val (P.8-321)」

Rmdir

フォルダを削除します。

・書式
Rmdir <directoryName>

・パラメータ

パラメータ名	データ型	説明
<directoryName>	文字列型	削除するフォルダのフォルダ名

・戻り値
なし

・解説
<directoryName> パラメータで指定したフォルダを削除します。
<directoryName> パラメータには、削除するフォルダのフォルダ名を絶対パスで指定してください。
次のような場合は、フォルダを削除せず Illegal function call のエラーが発生します。
- 指定したフォルダが存在しない場合
- フォルダの削除先に外部ストレージを指定した場合で、外部ストレージがセンサコントローラに未装着の場合
- 指定したフォルダに 1 つ以上のファイルが存在する場合

パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。

マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
E ドライブのルートにある「IMAGE2」フォルダを削除します。

```
Rmdir "E:¥IMAGE2"
```

・使用可能なモジュール
ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「Dskf (P.8-102)」
「Fcopy (P.8-115)」
「GetSystemData (P.8-133)」
「Isfile (P.8-154)」
「Kill (P.8-160)」
「Mkdir (P.8-188)」

RunOut

RUN 信号の出力状態を設定します。

・書式
RunOut <ioIdent>, <state>

・パラメータ

パラメータ名	データ型	説明
<ioIdent>	文字列型	使用する通信モジュールの識別名 （"Parallello" または "EtherCAT"） （参照：「9-4 IO モジュール一覧 (P.9-25)」）
<state>	整数型	端子の出力状態 0：出力 OFF 状態 1：出力 ON 状態

・戻り値
なし

・解説
<ioIdent> パラメータで指定した通信モジュールの RUN 信号に、<state> パラメータで指定した出力状態を設定します。
<ioIdent> パラメータには、通常は"Parallello" または"EtherCAT" を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
<ioIdent> パラメータとして、存在しない識別名を指定した場合は、Illegal function call のエラーが発生します。
<ioIdent> パラメータとして、" "（空の文字列を指定する）、または、<ioIdent> パラメータを省略した場合は、すべての通信モジュールが対象となります。
<state> パラメータとして、存在しない出力状態の値を指定しても、エラーは発生しません。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
通信コマンドマクロで、パラレル I/O の RUN 信号を ON に設定します。

```
IOMODULE$ = "Parallel0"

Rem 出力状態を設定します
RunOut IOMODULE$, 1
```

・使用可能なモジュール
シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 5.40 以降

・関連項目

「BusyOut (P.8-34)」
「GetAll (P.8-120)」
「GetPort (P.8-129)」
「JudgeOut (P.8-159)」
「PutAll (P.8-204)」
「PutPort (P.8-205)」

SaveBackupData

システム+ シーングループ 0 データを保存します。

・書式
SaveBackupData(<fileName>)

・パラメータ

パラメータ名	データ型	説明
<fileName>	文字列型	保存するシステム＋シーングループ 0 データファイル名（*.bkd）

・戻り値
なし

・解説
<fileName> パラメータで指定したファイル名のファイルに、システム+シーングループ 0 データを保存します。
<fileName> パラメータには、保存するファイルのファイル名を絶対パスで指定してください。
<fileName> パラメータに指定するファイル名には、ファイル拡張子（.bkd）を指定してください。
<fileName> パラメータで指定したファイルがすでに存在する場合は、ファイルが上書きされます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）
- FH シリーズの場合、保存先を RAMDisk および外部ストレージ以外のフォルダ（C:¥ProgramFiles¥FZ など）に保存しないでください。シーンデータ保存領域などが減少し、正しい動作ができなくなります。

保存先	FH シリーズ
RAMDisk	C:¥Data¥RAMDisk

保存先	FH シリーズ
外部ストレージ	E:¥、F:¥、G:¥、H:¥、M:¥、S:¥、T:¥、U:¥、V:¥、W:¥、X:¥、Y:¥、Z:¥

・使用例
システム＋シーングループ 0 データをファイルに保存します。

```
Rem システム＋シーングループ 0 データをファイルに保存します
SaveBackupData "E:¥BACKDIR¥BackupData.bkd"
```

・使用可能なモジュール
通信コマンドマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「GetSystemData (P.8-133)」
「LoadBackupData (P.8-168)」
「SaveData (P.8-227)」
「SaveScene (P.8-231)」
「SaveSceneGroup (P.8-232)」
「SaveSystemData (P.8-233)」
「SaveUnitData (P.8-234)」

SaveData

本体にデータを保存します。

・書式
SaveData

・パラメータ
なし

・戻り値
なし

・解説
センサコントローラ本体に、現在のシーングループデータとシステムデータを保存します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。(詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。)

・使用例
センサコントローラ本体に現在のシーングループデータとシステムデータを保存します。

```
Rem 本体にデータを保存します
SaveData
```

・使用可能なモジュール
通信コマンドマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「SaveBackupData (P.8-226)」
「SaveSceneGroup (P.8-232)」
「SaveUnitData (P.8-234)」
「SaveScene (P.8-231)」
「SaveSystemData (P.8-233)」

SaveImage

画像データを保存します。

・書式

SaveImage <preImageNo>, <fileName>

・パラメータ

パラメータ名	データ型	説明
<preImageNo>	整数型	保存する本体ロギング画像の番号（-1～すでにロギングしている本体ロギング画像の枚数 -1）
<fileName>	文字列型	保存する画像のファイル名

・戻り値

なし

・解説

<preImageNo> パラメータで指定した画像ロギング番号の画像を、<fileName> パラメータで指定した ifz 形式のファイル名で保存します。
<preImageNo> パラメータには、本体ロギング画像としてすでにロギングしているロギング画像の番号を指定してください。（詳細は、『画像処理システム FH/FHV シリーズ ユーザーズマニュアル (SDNB-719)』の「ロギング条件を設定する［ロギング設定］」を参照してください。）
<preImageNo> パラメータとして-1 を指定した場合は、最新入力画像を保存します。
<fileName> パラメータには、保存するファイルのファイル名を絶対パスで指定してください。
<fileName> パラメータに指定するファイル名には、ファイル拡張子（.ifz）を指定してください。
<fileName> パラメータで指定したファイルがすでに存在する場合は、ファイルが上書きされます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- ロギング設定にかかわらず、画像ファイルを IFZ ファイル形式で保存します。
- FH シリーズの場合、保存先を RAMDisk および外部ストレージ以外のフォルダ（C:¥ProgramFiles¥FZ など）に保存しないでください。シーンデータ保存領域などが減少し、正しい動作ができなくなります。

保存先	FH シリーズ
RAMDisk	C:¥Data¥RAMDisk
外部ストレージ	E:¥、F:¥、G:¥、H:¥、M:¥、S:¥、T:¥、U:¥、V:¥、W:¥、X:¥、Y:¥、Z:¥

・使用例

最新の計測画像をファイルに保存します。

```
Rem 最新入力画像をファイルとして保存します
SaveImage -1, "E:¥IMAGE¥sample.ifz"
```

・使用可能なモジュール
シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「GetSystemData (P.8-133)」　　「Remeasure (P.8-220)」

SaveMeasureImage

処理ユニットの計測画像を保存します。

・書式
SaveMeasureImage <measureImageNo>, <fileName>, <imageFormat>[, <startX>, <startY>, <sizeX>, <sizeY>]

・パラメータ

パラメータ名	データ型	説明
<measureImageNo>	整数型	保存対象の計測画像の計測画像番号（常に 0）
<fileName>	文字列型	保存する画像のファイル名
<imageFormat>	整数型	保存する画像の画像フォーマット 0：BMP 形式 10000～10100：JPEG 形式（10000 + JPEG 画像品質の 0～100） 20000～20009：PNG 形式（20000 + PNG 圧縮レベルの 0～9）
<startX>	整数型	保存対象とする画像領域の始点 X
<startY>	整数型	保存対象とする画像領域の始点 Y
<sizeX>	整数型	保存する画像の X 方向サイズ（1 以上）
<sizeY>	整数型	保存する画像の Y 方向サイズ（1 以上）

・戻り値
なし

・解説
<measureImageNo> パラメータで指定した計測画像を、<fileName> パラメータで指定したファイル名のファイルに、<imageFormat> パラメータで指定した画像フォーマットで、<startX> パラメータと<startY> パラメータで指定した左上を原点とするカメラ座標上の位置から、<sizeX> パラメータと<sizeY> パラメータで指定したピクセルサイズ分を切り出して保存します。
<startX> パラメータ、<startY> パラメータ、<sizeX> パラメータ、<sizeY> パラメータを省略した場合は、計測画像全体を保存します。
<measureImageNo> パラメータには常に 0 を指定して下さい。
<fileName> パラメータには、保存するファイルのファイル名を絶対パスで指定してください。
<fileName> パラメータに指定するファイル名には、ファイルの拡張子（.bmp など）をつけてください。
<fileName> パラメータで指定したファイルがすでに存在する場合は、ファイルが上書きされます。
<sizeX> パラメータと<sizeY> パラメータには、1 以上の値を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。

パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、*MEASUREPROC サブルーチンの中でしか使用できません。それ以外のサブルーチンで使用しようとした場合は、Illegal function call のエラーが発生します。
- FH シリーズの場合、保存先を RAMDisk および外部ストレージ以外のフォルダ（C:¥ProgramFiles¥FZ など）に保存しないでください。シーンデータ保存領域などが減少し、正しい動作ができなくなります。

保存先	FH シリーズ
RAMDisk	C:¥Data¥RAMDisk
外部ストレージ	E:¥、F:¥、G:¥、H:¥、M:¥、S:¥、T:¥、U:¥、V:¥、W:¥、X:¥、Y:¥、Z:¥

・使用例

計測画像 0 の画像全体を BMP 形式でファイルに保存します。

```
SaveMeasureImage 0, "E:¥IMAGE¥sample.bmp", 0
```

・使用可能なモジュール

ユニットマクロ

・対応バージョン

バージョン 4.00 以降

・関連項目

「GetImageSize (P.8-122)」
「GetSystemData (P.8-133)」
「GetUnitData (P.8-138)」

SaveScene

シーンデータを保存します。

・書式

SaveScene <sceneNo>, <fileName>

・パラメータ

パラメータ名	データ型	説明
<sceneNo>	整数型	保存するシーンのシーン番号（0～127）
<fileName>	文字列型	保存するシーンデータファイル名（*.scn）

・戻り値

なし

・解説

<fileName> パラメータで指定したファイル名のファイルに、<sceneNo> パラメータで指定したシーン番号のシーンデータを保存します。
<fileName> パラメータには、保存するファイルのファイル名を絶対パスで指定してください。
<fileName> パラメータに指定するファイル名には、ファイル拡張子（.scn）を指定してください。
<fileName> パラメータで指定したファイルがすでに存在する場合は、ファイルが上書きされます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）
- FH シリーズの場合、保存先を RAMDisk および外部ストレージ以外のフォルダ（C:¥ProgramFiles¥FZ など）に保存しないでください。シーンデータ保存領域などが減少し、正しい動作ができなくなります。

保存先	FH シリーズ
RAMDisk	C:¥Data¥RAMDisk
外部ストレージ	E:¥、F:¥、G:¥、H:¥、M:¥、S:¥、T:¥、U:¥、V:¥、W:¥、X:¥、Y:¥、Z:¥

・使用例

シーン番号 2 のシーンデータをファイルに保存します。

```
Rem シーン番号 2 のシーンデータをファイルに保存します
SaveScene 2, "E:¥BACKDIR¥scene02.scn"
```

・使用可能なモジュール

通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「GetSystemData (P.8-133)」　「LoadScene (P.8-169)」

「LoadBackupData (P.8-168)」
「SaveSceneGroup (P.8-232)」
「SaveUnitData (P.8-234)」
「SaveData (P.8-227)」
「SaveSystemData (P.8-233)」
「SceneNo (P.8-241)」

SaveSceneGroup

シーングループデータを保存します。

・書式

SaveSceneGroup <sceneGroupNo>, <fileName>

・パラメータ

パラメータ名	データ型	説明
<sceneGroupNo>	整数型	保存するシーングループのシーンシーングループ番号（0～31）
<fileName>	文字列型	保存するシーングループデータファイル名（*.sgp）

・戻り値

なし

・解説

<fileName> パラメータで指定したファイル名のファイルに、<sceneNo> パラメータで指定したシーングループ番号のシーングループデータを保存します。
<fileName> パラメータには、保存するファイルのファイル名を絶対パスで指定してください。
<fileName> パラメータに指定するファイル名には、ファイル拡張子（.sgp）を指定してください。
<fileName> パラメータで指定したファイルがすでに存在する場合は、ファイルが上書きされます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）
- FH シリーズの場合、保存先を RAMDisk および外部ストレージ以外のフォルダ（C:¥ProgramFiles¥FZ など）に保存しないでください。シーンデータ保存領域などが減少し、正しい動作ができなくなります。

保存先	FH シリーズ
RAMDisk	C:¥Data¥RAMDisk
外部ストレージ	E:¥、F:¥、G:¥、H:¥、M:¥、S:¥、T:¥、U:¥、V:¥、W:¥、X:¥、Y:¥、Z:¥

・使用例

シーングループ番号 2 のシーングループデータをファイルに保存します。

```
Rem シーングループ番号 2 のシーングループデータをファイルに保存します
SaveSceneGroup 2, "E:¥BACKDIR¥scenegroup02.sgp"
```

・使用可能なモジュール

通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「GetSystemData (P.8-133)」
「LoadBackupData (P.8-168)」
「SaveScene (P.8-231)」
「SaveUnitData (P.8-234)」
「LoadSceneGroup (P.8-170)」
「SaveData (P.8-227)」
「SaveSystemData (P.8-233)」
「SceneGroupNo (P.8-238)」

SaveSystemData

システムデータを保存します。

・書式

SaveSystemData <fileName>

・パラメータ

パラメータ名	データ型	説明
<fileName>	文字列型	保存するシステムデータファイル名（*.ini）

・戻り値

なし

・解説

<fileName> パラメータで指定したファイル名のファイルに、システムデータを保存します。
<fileName> パラメータには、保存するファイルのファイル名を絶対パスで指定してください。
<fileName> パラメータに指定するファイル名には、ファイル拡張子（.ini）を指定してください。
<fileName> パラメータで指定したファイルがすでに存在する場合は、ファイルが上書きされます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）
- FH シリーズの場合、保存先を RAMDisk および外部ストレージ以外のフォルダ（C:¥ProgramFiles¥FZ など）に保存しないでください。シーンデータ保存領域などが減少し、正しい動作ができなくなります。

保存先	FH シリーズ
RAMDisk	C:¥Data¥RAMDisk
外部ストレージ	E:¥、F:¥、G:¥、H:¥、M:¥、S:¥、T:¥、U:¥、V:¥、W:¥、X:¥、Y:¥、Z:¥

・使用例

システムデータをファイルに保存します。

```
Rem システムデータをファイルに保存します
SaveSystemData "E:¥BACKDIR¥backupsysset.ini"
```

・使用可能なモジュール

通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「GetSystemData (P.8-133)」
「LoadBackupData (P.8-168)」
「SaveScene (P.8-231)」
「SaveUnitData (P.8-234)」
「LoadSystemData (P.8-171)」
「SaveData (P.8-227)」
「SaveSceneGroup (P.8-232)」

SaveUnitData

処理ユニットデータを保存します。

・書式

SaveUnitData <sceneNo>, <unitNo>, <unitCount>, <fileName>

・パラメータ

パラメータ名	データ型	説明
<sceneNo>	整数型	保存するシーンのシーン番号（-1～127）
<unitNo>	整数型	保存を開始する処理ユニット番号（0～（現在シーンの処理ユニット数 -1））
<unitCount>	整数型	保存する処理ユニット数（-1、1～（現在シーンの処理ユニット数 - 保存を開始する処理ユニット番号））
<fileName>	文字列型	保存する処理ユニットデータファイル名（*.unt）

・戻り値

なし

・解説

<sceneNo> パラメータで指定したシーン番号の、<unitNo> パラメータで指定した処理ユニット番号の処理ユニットから<unitCount> パラメータで指定した処理ユニット数だけ、<fileName> パラメータで指定したファイル名のファイルに、処理ユニットデータを保存します。
<sceneNo> パラメータに-1 を指定した場合は、保存するシーンのシーン番号には現在のシーンのシーン番号が指定されます。
<unitCount> パラメータに-1 を指定した場合は、処理ユニットデータファイルに含まれるすべての処理ユニットデータが保存されます。
<fileName> パラメータには、保存するファイルのファイル名を絶対パスで指定してください。
<fileName> パラメータに指定するファイル名には、ファイル拡張子（.unt）を指定してください。
<fileName> パラメータで指定したファイルがすでに存在する場合は、ファイルが上書きされます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）
- FH シリーズの場合、保存先を RAMDisk および外部ストレージ以外のフォルダ（C:¥ProgramFiles¥FZ など）に保存しないでください。シーンデータ保存領域などが減少し、正しい動作ができなくなります。

保存先	FH シリーズ
RAMDisk	C:¥Data¥RAMDisk
外部ストレージ	E:¥、F:¥、G:¥、H:¥、M:¥、S:¥、T:¥、U:¥、V:¥、W:¥、X:¥、Y:¥、Z:¥

・使用例
現在のシーンの処理ユニット番号 2 から処理ユニット番号 4 の処理ユニットを、処理ユニットデータファイルに保存します。

```
Rem 現在のシーンの処理ユニット 2 から処理ユニット 4 を処理ユニットデータファイルに保存します
SaveUnitData -1, 2, 3, "E:¥BACKDIR¥unitsave.unt"
```

・使用可能なモジュール
通信コマンドマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「GetSystemData (P.8-133)」
「SaveBackupData (P.8-226)」
「SaveScene (P.8-231)」
「SaveSystemData (P.8-233)」
「UnitNo (P.8-318)」
「LoadUnitData (P.8-172)」
「SaveData (P.8-227)」
「SaveSceneGroup (P.8-232)」
「SceneNo (P.8-241)」
「Ut (P.8-320)」

SceneCount

使用可能なシーン数を取得します。

・書式
SceneCount

・パラメータ
なし

・戻り値
整数型の使用可能なシーン数の値を返します。

・解説
使用可能なシーン数を取得します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
使用可能なシーン数を取得します。

```
NUM& = SceneCount
```

・使用可能なモジュール
シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「ChangeScene (P.8-38)」
「CopyScene (P.8-52)」
「ClearScene (P.8-43)」
「SceneNo (P.8-241)」

SceneDescription$

シーンの説明を取得します。

・書式

SceneDescription$(<sceneNo>)

・パラメータ

パラメータ名	データ型	説明
<sceneNo>	整数型	説明を取得するシーンのシーン番号（0～127）

・戻り値

文字列型のシーンの説明の値を返します。

・解説

<sceneNo> パラメータで指定したシーン番号のシーンに設定された説明を取得します。
説明が未設定の場合は、ヌル文字列("") を返します。
シーンの説明は、SetSceneDescription 関数を実行するか、シーンメンテナンス画面で設定することができます。（参照：「SetSceneDescription (P.8-265)」）、（詳細は、『画像処理システム FH/FHV シリーズ ユーザーズマニュアル (SDNB-719)』の「シーンを編集する」を参照してください。）
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

シーン 1 の説明を取得して、説明が未設定の場合には説明を設定します。

```
Rem シーンの説明を取得します
DESCRIPTION$ = SceneDescription$(1)

If DESCRIPTION$ = "" Then
   Rem シーンの説明を設定します
   SetSceneDescription 1, "Description 1"
Endif
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「SceneMaker$ (P.8-240)」
「SceneTitle$ (P.8-242)」
「SetSceneDescription (P.8-265)」
「SetSceneMaker (P.8-268)」
「SetSceneTitle (P.8-269)」

SceneGroupCount

使用可能なシーングループ数を取得します。

・書式
SceneGroupCount

・パラメータ
なし

・戻り値
整数型の使用可能なシーングループ数の値を返します。

・解説
使用可能なシーングループ数を取得します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
使用可能なシーングループ数を取得します。

```
NUM& = SceneGroupCount
```

・使用可能なモジュール
シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「ChangeSceneGroup (P.8-39)」
「ClearSceneGroup (P.8-44)」
「CopySceneGroup (P.8-53)」
「SceneGroupNo (P.8-238)」

SceneGroupNo

現在のシーングループのシーングループ番号を取得します。

・書式

SceneGroupNo

・パラメータ

なし

・戻り値

整数型の現在のシーングループのシーングループ番号の値を返します。

・解説

現在のシーングループのシーングループ番号を取得します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

現在のシーングループのシーングループ番号を取得して、シーングループ番号が 2 であればシーン 3 に切り替えます。

```
Rem 現在のシーングループのシーングループ番号を取得します
NO& = SceneGroupNo

Rem シーングループ番号が 2 であればシーン 3 に切り替えます
If NO&＝ 2 Then
   ChangeScene 3
Endif
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「ChangeScene (P.8-38)」
「ChangeSceneGroup (P.8-39)」
「ClearSceneGroup (P.8-44)」
「CopySceneGroup (P.8-53)」
「SceneGroupCount (P.8-237)」
「SceneNo (P.8-241)」

SceneGroupTitle$

シーングループのタイトル名を取得します。

・書式

SceneGroupTitle$(<sceneGroupNo>)

・パラメータ

パラメータ名	データ型	説明
<sceneGroupNo>	整数型	シーングループタイトル名を取得するシーングループのシーングループ番号（0～31）

・戻り値

文字列型のシーングループのタイトル名の値を返します。

・解説

<sceneGroupNo> パラメータで指定したシーングループ番号のシーングループに設定されたタイトル名を取得します。
タイトル名が未設定の場合は、"シーングループ 0" などの既定の文字列を返します。
シーングループのタイトル名は、SetSceneGroupTitle 関数を実行するか、シーンメンテナンス画面で設定することができます。（参照：「SetSceneGroupTitle (P.8-267)」）、（詳細は、『画像処理システム FH/FHV シリーズ ユーザーズマニュアル (SDNB-719)』の「シーンを編集する」を参照してください。）
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

シーングループ 2 のタイトル名を取得します。

```
TITLE$ = SceneGroupTitle$(2)
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「SceneTitle$ (P.8-242)」
「SetSceneTitle (P.8-269)」

SceneMaker$

シーンの作成者名を取得します。

・書式

SceneMaker$(<sceneNo>)

・パラメータ

パラメータ名	データ型	説明
<sceneNo>	整数型	作成者名を取得するシーンのシーン番号（0～127）

・戻り値

文字列型のシーンの作成者名の値を返します。

・解説

<sceneNo> パラメータで指定したシーン番号のシーンに設定された作成者名を取得します。
作成者名が未設定の場合は、ヌル文字列("") を返します。
シーンの作成者名は、SetSceneMaker 関数を実行するか、シーンメンテナンス画面で設定することができます。（参照：「SetSceneMaker (P.8-268)」）、（詳細は、『画像処理システム FH/FHV シリーズ ユーザーズマニュアル (SDNB-719)』の「シーンを編集する」を参照してください。）
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

シーン 3 の作成者名を取得して、未設定であれば作成者名を設定します。

```
Rem シーンの作成者名を取得します
NAME$ = SceneMaker$(3)

If NAME$ = "" Then
   Rem シーンの作成者名を設定します
   SetSceneMaker 3, "Maker"
Endif
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「SceneDescription$ (P.8-236)」
「SetSceneDescription (P.8-265)」
「SetSceneTitle (P.8-269)」
「SceneTitle$ (P.8-242)」
「SetSceneMaker (P.8-268)」

SceneNo

現在のシーンのシーン番号を取得します。

・書式
SceneNo

・パラメータ
なし

・戻り値
整数型の現在のシーンのシーン番号の値を返します。

・解説
現在のシーンのシーン番号を取得します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
現在のシーンのシーン番号を取得して、シーン番号が 2 でなければシーン 2 に切り替えます。

```
Rem 現在のシーンのシーン番号を取得します
NO& = SceneNo

Rem シーン番号が 2 でなければシーン 2 に切り替えます
If NO& <> 2 Then
   ChangeScene 2
Endif
```

・使用可能なモジュール
通信コマンドマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「ChangeScene (P.8-38)」
「ClearScene (P.8-43)」
「CopyScene (P.8-52)」
「SceneCount (P.8-235)」

SceneTitle$

シーンのタイトル名を取得します。

・書式

SceneTitle$(<sceneNo>)

・パラメータ

パラメータ名	データ型	説明
<sceneNo>	整数型	シーンタイトル名を取得するシーンのシーン番号（0～127）

・戻り値

文字列型のシーンのタイトル名の値を返します。

・解説

<sceneNo> パラメータで指定したシーン番号のシーンに設定されたタイトル名を取得します。
タイトル名が未設定の場合は、"シーン 0" などの既定の文字列を返します。
シーンのタイトル名は、SetSceneTitle 関数を実行するか、シーンメンテナンス画面またはフロー編集画面で設定することができます。（参照：「SetSceneTitle (P.8-269)」）、（詳細は、『画像処理システム FH/FHV シリーズ ユーザーズマニュアル (SDNB-719)』の「シーンを編集する」、「シーン内の処理ユニットを編集する」を参照してください。）
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

シーン 2 のタイトル名を取得します。

```
TITLE$ = SceneTitle$(2)
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「SceneDescription$ (P.8-236)」
「SetSceneDescription (P.8-265)」
「SetSceneTitle (P.8-269)」
「SceneMaker$ (P.8-240)」
「SetSceneMaker (P.8-268)」

ScreenCapture

画面のキャプチャを保存します。

・書式

ScreenCapture <fileName>

・パラメータ

パラメータ名	データ型	説明
<fileName>	文字列型	画面キャプチャを保存するファイル名

・戻り値

なし

・解説

センサコントローラの画面をキャプチャして、<fileName> パラメータで指定した BMP 形式のファイル名で保存します。
<fileName> パラメータには、保存するファイルのファイル名を絶対パスで指定してください。
<fileName> パラメータには、BMP 形式のファイルとして拡張子（.bmp）のファイル名を指定してください。"bmp" 以外の拡張子を指定した場合は、画面を正しくキャプチャできません。
<fileName> パラメータで指定したファイルがすでに存在する場合は、ファイルが上書きされます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- FH シリーズの場合、保存先を RAMDisk および外部ストレージ以外のフォルダ（C:¥ProgramFiles¥FZ など）に保存しないでください。シーンデータ保存領域などが減少し、正しい動作ができなくなります。

保存先	FH シリーズ
RAMDisk	C:¥Data¥RAMDisk
外部ストレージ	E:¥、F:¥、G:¥、H:¥、M:¥、S:¥、T:¥、U:¥、V:¥、W:¥、X:¥、Y:¥、Z:¥

・使用例

画面をキャプチャして、ファイル名"E:¥IMAGE¥samplecapture.bmp" のファイルに保存します。

```
ScreenCapture "E:¥IMAGE¥samplecapture.bmp"
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「GetSystemData (P.8-133)」
「Str$ (P.8-287)」
「Str2$ (P.8-288)」

Select Case Case Else End Select

指定した条件によって処理の流れを制御します。

・書式

Select <expression>
[Case <value>
<caseStatement>]
:
:
[Case Else
<elseStatement>]
End Select

・パラメータ

パラメータ名	データ型	説明
<expression>	整数型	処理の流れを制御する数式
<value>	整数型	数式の値と比較する数値
<caseStatement>	-	数式と数値が一致したときに実行するステートメント
<elseStatement>	-	すべての数値が数式と一致しなかったときに実行するステートメント

・戻り値

なし

・解説

<expression> パラメータで指定した数式に対して、複数の Case ブロックの中から<value> パラメータで指定した数値が数式と一致した Case ブロックのステートメントを実行します。
<expression> パラメータで指定した数式に対して、すべての<value> パラメータで指定した数値が一致しなかった場合は、<elseStatement> パラメータで指定した Case Else ブロックのステートメントを実行します。
複数の<value> パラメータで指定した数値が数式と一致する場合は、Select 文の数式から順に最初に一致した数値の Case ブロック内のステートメントだけを実行します。
Case ブロックと Case Else ブロックは省略可能です。
ステートメントの中で Goto 関数を使用して Case ブロックまたは Case Else ブロックの中から外に制御を移したり、外から中に制御を移したりした場合は、予期せぬ動作になる可能性があります。
Select 文と End Select 文のいずれかまたは両方を記述しなかった場合は、記述の内容によって CASE without SELECT、END SELECT without SELECT、SELECT without END SELECT、CASE without END SELECT のいずれかのエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

ユニットマクロ処理ユニットの*MEASUREDISPG サブルーチンの中で、メイン画面の画像表示ウィンドウに設定されている画像表示サブ番号によって、画像表示ウィンドウにグラフィック表示する内容を変えます。

```
*MEASUREDISPG

   Rem 表示サブ番号を取得します
   SUBNO& = DisplaySubNo

   Rem 表示サブ番号によって画像表示ウィンドウに表示する内容を変えます
   Select SUBNO&
   Case 1
      Rem 表示サブ番号が 1 のときは処理ユニット 1 のタイトルを判定結果に応じた色で表示します
      SetTextStyle 24, TA_LEFT, UnitJudge(1), 0, FONTSTYLE_NORMAL
      TEXT$ = UnitTitle$(1)
   Case 2
      Rem 表示サブ番号が 2 のときは処理ユニット 2 のタイトルを判定結果に応じた色で表示します
      SetTextStyle 24, TA_LEFT, UnitJudge(2), 0, FONTSTYLE_NORMAL
      TEXT$ = UnitTitle$(2)
   Case Else
      Rem 表示サブ番号が 1、2 以外のときは未計測色で表示します
      SetTextStyle 24, TA_LEFT, JUDGE_NC, 0, FONTSTYLE_NORMAL
      TEXT$ = "Error"
   End Select

   Rem 画像表示ウィンドウにテキストを表示します
   DrawTextG TEXT$, 50, 0, 0, UnitNo

Return
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「DisplaySubNo (P.8-69)」
「Gosub (P.8-140)」
「If Then Elseif Else Endlf (P.8-144)」
「UnitJudge (P.8-317)」
「DrawTextG (P.8-99)」
「If Then Else (P.8-143)」
「SetTextStyle (P.8-272)」
「UnitNo (P.8-318)」

SendData

データを送信します。

・書式

SendData <ioIdent>, <outputData()>, <outputSize>[, <parameter()>, <parameterSize>]

・パラメータ

パラメータ名	データ型	説明
<ioIdent>	文字列型	使用する通信モジュールの識別名 （参照：「9-4 IO モジュール一覧 (P.9-25)」）
<outputData()>	整数型配列	送信するデータ
<outputSize>	整数型	送信するデータのサイズ
<parameter()>	整数型配列	オプションで指定するパラメータデータ
<parameterSize>	整数型	オプションで指定するパラメータデータのサイズ

・戻り値

なし

・解説

<ioIdent> パラメータで指定した通信モジュールを使用して、<outputData()> パラメータで指定したデータを<outputSize> パラメータで指定したサイズ分送信します。
<outputData()> パラメータには、送信するデータを格納した 1 次元の整数型配列変数を、要素番号を指定せず() だけ付加して指定してください。無手順 UDP 通信、無手順 TCP 通信ともにデリミタは不要です。
<parameter()> パラメータと<parameterSize> パラメータに設定できる値は、<ioIdent> パラメータで指定した通信モジュールによって異なります。詳細は「9-4 IO モジュール一覧 (P.9-25)」をご覧ください。<parameter()> パラメータと<parameterSize> パラメータは省略可能です。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
ハンドシェイクありの設定で通信がタイムアウトした場合などデータ送信に失敗した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

無手順 UDP 通信で、送信先を指定してデータを送信します。

```
Rem 送信先アドレス情報を作成します(10.1.1.101)
Dim IPADDR&(4)
IPADDR&(0) = 10
IPADDR&(1) = 1
IPADDR&(2) = 1
IPADDR&(3) = 101

Rem 送信するデータを作成します
Dim BUFFER&(4)
BUFFER&(0) = 1
BUFFER&(1) = 2
BUFFER&(2) = 3
BUFFER&(3) = 4
BUFFER&(4) = 5

Rem アドレスを指定してデータを送信します
SendData "UdpNormal", BUFFER&(), 4 * 5, IPADDR&(), 4 * 4
```

PLC リンク通信で、オフセット値を指定してデータ出力エリアにデータを書き込みます。

```
Rem オフセットデータを作成します
Dim OFFSET&(0)
OFFSET&(0) = 2

Rem 送信するデータを作成します
Dim BUFFER&(4)
BUFFER&(0) = 1
BUFFER&(1) = 2
BUFFER&(2) = 3
BUFFER&(3) = 4
BUFFER&(4) = 5

Rem オフセット値を使用してデータを送信します
SendData "SerialPlcLink", BUFFER&(), 4 * 5, OFFSET&(), 4 * 1
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「ReceiveData (P.8-212)」　「SendString (P.8-248)」

SendString

文字列データを送信します。

・書式

SendString <ioIdent>, <outputString>

・パラメータ

パラメータ名	データ型	説明
<ioIdent>	文字列型	使用する通信モジュールの識別名 （参照：「9-4 IO モジュール一覧 (P.9-25)」）
<outputString>	文字列型	送信する文字列

・戻り値

なし

・解説

<ioIdent> パラメータで指定した通信モジュールを使用して、<outputString> パラメータで指定した文字列を送信します。無手順 UDP ではデリミタは不要です。無手順 TCP ではデリミタ CR が必要です。デリミタは変更できません。

このマクロ関数に対応していない通信モジュールがあります。（参照：「9-4 IO モジュール一覧 (P.9-25)」）

パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。

パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定しても、Illegal function call のエラーは発生しません。

文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。

マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- UDP 無手順通信を使用したシーン制御マクロまたはユニットマクロ処理の場合で PLC などの外部装置から一度もコマンド受信をしていないとき、SendString を実行すると Illegal function call のエラーが発生します。
- 設定できる文字列は、以下のとおりです。
  ·形 FH シリーズの場合：「Language setting」で設定した言語と英語が設定可能です。

・使用例

TCP 無手順通信で文字列を送信します。

```
Rem 送信する文字列を作成します
DATA$ = "Test string"

Rem 文字列を送信します
SendString "TcpNormal", DATA$
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「ReceiveData (P.8-212)」　「SendData (P.8-246)」

SetDisplayUnitNo

フロー表示ウィンドウの処理ユニット番号の選択状態を設定します。

・書式

SetDisplayUnitNo <unitNo>

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	選択状態にする処理ユニット番号

・戻り値

なし

・解説

<unitNo> パラメータで指定した処理ユニット番号の処理ユニットを、フロー表示ウィンドウ上で選択状態にします。
画像表示ウィンドウとテキスト表示ウィンドウの表示対象の処理ユニットで「フロー表示に連動」を設定している場合に、フロー表示ウィンドウ上で選択状態になった処理ユニットの情報が表示されます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定しても、Illegal function call のエラーは発生しません。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

通信コマンドマクロにおいてコマンドの引数で指定したユニットをフロー表示ウィンドウで選択状態にします。

```
Rem 通信コマンドの引数で指定された番号の処理ユニットを選択します
SetDisplayUnitNo argumentValue#(0)
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「DisplayUnitNo (P.8-70)」
「UnitNo (P.8-318)」
「SetImageWindow (P.8-254)」
「Ut (P.8-320)」

SetDrawStyle

グラフィック図形の描画属性を設定します。

・書式

SetDrawStyle <style>, <width>, <color>

・パラメータ

パラメータ名	データ型	説明
<style>	整数型	描画するグラフィックの線の種類 PS_SOLID：実線 PS_DASH：破線*1 PS_DOT：点線*1 PS_DASHDOT：一点鎖線*1 PS_DASHDOTDOT：二点鎖線*1 PS_NULL：線なし PS_INSIDEFRAME：実線*2
<width>	整数型	描画するグラフィックの線の幅
<color>	整数型	描画するグラフィックの線の色のカラー値 JUDGE_NC：未計測色（灰色） JUDGE_OK：OK 色（緑色） JUDGE_NG：NG 色（赤色） RGB 関数：任意の色

*1. 線の幅が 1 のときのみ有効

*2. 下記の図形のときのみ有効

- 幅付円形
- 楕円
- 幅付円弧

・戻り値

なし

・解説

<style> パラメータで指定した線の種類と、<width> パラメータで指定した線の幅と、<color> パラメータで指定した線の色を描画属性として設定します。画像表示ウィンドウ制御のグラフィック図形を描画するマクロ関数を実行する前にあらかじめこのマクロ関数を実行することで、設定した描画属性のグラフィック図形を描画することができます。DrawTextG 関数の描画属性を設定する場合は、SetTextStyle 関数を使用してください。（参照：「SetTextStyle (P.8-272)」）

<style> パラメータで PS_DASH、PS_DASHDOT、PS_DASHDOTDOT のいずれかを指定した場合は、<width> パラメータを 1 に設定してください。<width> パラメータが 1 以外の場合は、グラフィックを実線で描画します。

<style> パラメータで PS_INSIDEFRAME を指定した場合は、円形／幅付き円形／楕円／円弧／幅付き円弧を描画するときに<width> パラメータで指定した線の幅が考慮され、その図形に収まるように図形を縮小します。これら以外の図形を描画する場合は、PS_SOLID と同じ線の種類で描画します。

<color> パラメータとして、RGB 関数で取得したカラー値を設定することもできます。（参照：「RGB (P.8-222)」）

パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。

整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。

マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、*MEASUREDISPI サブルーチンまたは*MEASUREDISPG サブルーチンの中でしか使用できません。それ以外のサブルーチンで使用しようとした場合は、Illegal function call のエラーが発生します。

・使用例

ユニットマクロ処理ユニットの*MEASUREDISPG サブルーチンの中で、線の種類が破線、線の色がOK 色の直線をグラフィック表示します。

```
*MEASUREDISPG

    Rem 描画属性を設定します
    SetDrawStyle PS_DASH, 1, JUDGE_OK

    Rem グラフィックを描画します
    DrawLine 100, 100, 500, 400, 0, UnitNo

Return
```

・使用可能なモジュール

ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「DrawArc (P.8-74)」
「DrawArcW (P.8-75)」
「DrawBox (P.8-77)」
「DrawCircle (P.8-79)」
「DrawCircleW (P.8-80)」
「DrawCursor (P.8-82)」
「DrawEllipse (P.8-83)」
「DrawFigure (P.8-85)」
「DrawLine (P.8-88)」
「DrawLineW (P.8-90)」
「DrawPoint (P.8-92)」
「DrawPolygon (P.8-93)」
「DrawSearchFigure (P.8-95)」
「RGB (P.8-222)」
「SetTextStyle (P.8-272)」

SetForegroundLine

複数ラインランダムトリガモードまたはノンストップ調整モードで、操作対象のラインを指定します。

・書式

SetForegroundLine <lineNo>

・パラメータ

パラメータ名	データ型	説明
<lineNo>	整数型	操作対象のライン番号（0～7） ノンストップ調整モードのライン番号は、次のとおりです。 • 0：計測稼動側 • 1：ノンストップ調整側

・戻り値

なし

・解説

<lineNo> パラメータで、操作対象のライン番号を指定します。指定したライン番号の画面が前面に表示されます。

操作対象のライン番号が起動している状態で実行してください。起動していない場合は、Illegal function call のエラーが発生します。

パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。

・使用上の注意

なし

・使用例

```
Rem 無手順で表示するラインを切り替える
NO& = int(Argumentvalue#(0))
SetForegroundLine NO&
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 5.40 以降

・関連項目

なし

SetGlobalData

グローバルデータを設定します。

・書式

SetGlobalData <dataIdent>, <data>

・パラメータ

パラメータ名	データ型	説明
<dataIdent>	文字列型	値を設定するグローバルデータの識別名
<data>	整数型 実数型 文字列型	グローバルデータに設定する値

・戻り値

なし

・解説

<dataIdent> パラメータで指定した識別名のグローバルデータに、<data> パラメータで指定した値を設定します。
指定した識別名のグローバルデータが存在しない場合は、<dataIdent> パラメータで指定した識別名のグローバルデータを追加して、追加したデータに<data> パラメータで指定した値を設定します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
<dataIdent> パラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。(詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。)

・使用例

識別名が"ABC" のグローバルデータに、値として 1 を設定します。

```
Rem グローバルデータ"ABC" の値に 1 を設定します
SetGlobalData "ABC", 1

Rem グローバルデータ"ABC" に設定された値（整数値）を取得し、変数 DATA& に格納します
GetGlobalData "ABC", DATA&
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「AddGlobalData (P.8-23)」
「GetGlobalData (P.8-121)」

SetImageWindow

画像表示ウィンドウの状態を設定します。

・書式

[シーン制御マクロ ／ 通信コマンドマクロ]

SetImageWindow <windowNo>, <locationX>, <locationY>, <width>, <height>, <unitNo>, <subNo>, <magnification>, <originX>, <originY>, <update>, <visible>

[ユニットマクロ]

SetImageWindow <magnification>, <originX>, <originY>

・パラメータ

パラメータ名	データ型	説明
<windowNo>	整数型	状態を設定する画像表示ウィンドウの番号（0～23）
<locationX>	整数型	画像表示ウィンドウの左上 X 座標
<locationY>	整数型	画像表示ウィンドウの左上 Y 座標
<width>	整数型	画像表示ウィンドウの幅
<height>	整数型	画像表示ウィンドウの高さ
<unitNo>	整数型	表示対象の処理ユニットの処理ユニット番号（-1～（現在シーンの処理ユニット数 -1）
<subNo>	整数型	表示対象のサブ画像番号（-1～100）
<magnification>	実数型	表示倍率（-1、1～16）
<originX>	整数型	画像表示ウィンドウの左上座標に対する表示画像の左上 X 座標
<originY>	整数型	画像表示ウィンドウの左上座標に対する表示画像の左上 Y 座標
<update>	整数型	画像モード、画像ウィンドウの更新タイミング 0：計測時毎回（カメラ画像フリーズ） 1：計測時総合判定結果 NG の場合のみ（最新 NG 画像） 2：計測時対象処理ユニット NG の場合のみ 3：常時更新（カメラ画像スルー）
<visible>	整数型	表示有無 0：ウィンドウ非表示 1：ウィンドウ表示

・戻り値

なし

・解説

<windowNo> パラメータで指定した画像表示ウィンドウの状態を設定します。このマクロ関数をユニットマクロで使用する場合は、MEASUREDISPI サブルーチンで表示処理を実行した画像表示ウィンドウが対象になります。

<locationX> パラメータと<locationY> パラメータには、画像コンテナウィンドウの左上座標から画像表示ウィンドウの左上座標への相対座標の値を指定してください。

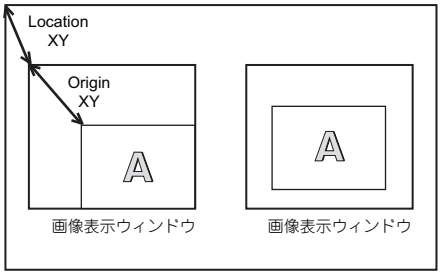
<width> パラメータと<height> パラメータには、画像表示ウィンドウの幅と高さの値を設定してください。

<unitNo> パラメータには、表示対象の処理ユニット番号の値を指定してください。画像表示ウィンドウの表示対象をフロー表示に連動させる場合は、-1 を指定してください。

<subNo> パラメータには、表示対象のサブ画像番号の値を指定してください。画像表示ウィンドウの表示内容を位置一覧表示にする場合は、-1 を指定してください。

<magnification> パラメータには、画像表示ウィンドウの表示倍率の値を指定してください。0 は設定しないでください。表示倍率を自動にする場合は、-1 を指定してください。

<originX> パラメータと<originY> パラメータには、画像ウィンドウの左上座標から表示画像の左上座標への相対座標の値を指定してください。



<update> パラメータには、画像表示ウィンドウの画像モードの値を指定してください。
<visible> パラメータには、画像表示ウィンドウの表示状態の値を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定しても、Illegal function call のエラーは発生しません。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

[シーン制御マクロ ／ 通信コマンドマクロ]

- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。(詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。)

[ユニットマクロ]

- このマクロ関数は、*MEASUREDISPI サブルーチンまたは*MEASUREDISPG サブルーチンの中でしか使用できません。それ以外のサブルーチンで使用しようとした場合は、Illegal function call のエラーが発生します。

・使用例

通信コマンドマクロで、画像表示ウィンドウ 0～3 の画像モードをスルー表示に変更します。通信コマンドマクロでは、あらかじめ BusyOn フラグを ON の設定にしておきます。

```
For I& = 0 To 3
   Rem 画像表示ウィンドウの状態を取得します
   GetImageWindow I&, LOCATIONX&, LOCATIONY&, WIDTH&, HEIGHT&, UNITNO&, SUBNO&, MAG#, ORIGINX&, ORI
GINY&, UPDATE&, VISIBLE&

   Rem 更新タイミングをスルーに変更します
   UPDATE& = 3

   Rem 画像表示ウィンドウの状態を設定します
   SetImageWindow I&, LOCATIONX&, LOCATIONY&, WIDTH&, HEIGHT&, UNITNO&, SUBNO&, MAG#, ORIGINX&, ORI
GINY&, UPDATE&, VISIBLE&

Next
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「DisplayUnitNo (P.8-70)」
「GetImageWindow (P.8-123)」
「GetTextWindow (P.8-137)」
「SetDisplayUnitNo (P.8-249)」
「UnitNo (P.8-318)」

SetMeasureImage

処理ユニットの計測画像を設定します。

・書式

SetMeasureImage <measureImageNo>, <unitNo>, <imageNo>

・パラメータ

パラメータ名	データ型	説明
<measureImageNo>	整数型	設定対象の計測画像の計測画像番号（常に 0）
<unitNo>	整数型	計測画像として設定する画像を持つ処理ユニットの処理ユニット番号
<ImageNo>	整数型	計測画像として設定する画像の画像番号

・戻り値

なし

・解説

<measureImageNo> パラメータで指定した計測画像として、<unitNo> パラメータで指定した処理ユニットが持つ<imageNo> パラメータで指定した画像番号の画像を設定します。計測画像は、計測フロー上の後続の処理ユニットが計測や画像処理に使用できる画像です。
<measureImageNo> パラメータには、通常は 0 を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、*MEASUREPROC サブルーチンの中でしか使用できません。それ以外のサブルーチンで使用しようとした場合は、マクロ関数の処理は実行されずエラーも発生しません。

・使用例

後続の処理ユニットが使用する計測画像を処理ユニット番号 4 のカメラ切替の画像に変更します。

```
Rem 計測画像として処理ユニット番号 4 のカメラ切替の計測画像 0 を設定します
SetMeasureImage 0, 4, 0
```

・使用可能なモジュール

ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「CopyMeasureImage (P.8-51)」
「UnitNo (P.8-318)」
「CopyUnitImage (P.8-56)」
「Ut (P.8-320)」

SetMeasureOut

計測結果の外部出力設定を設定します。

・書式

SetMeasureOut <mode>

・パラメータ

パラメータ名	データ型	説明
<mode>	整数型	外部出力設定 0：外部に出力しない 1：外部に出力する

・戻り値

なし

・解説

<mode> パラメータで指定した外部出力設定の値であるレイアウト設定の「外部に出力する」の値を設定します。(詳細は、『画像処理システム FH/FHV シリーズ ユーザーズマニュアル (SDNB-719)』の「レイアウトごとの出力信号の動作をアレンジする［レイアウト設定］」を参照してください。)
計測結果の外部出力設定の値に 0 を設定した場合でも、マクロカスタマイズ機能で SendData 関数や SendString 関数を使用した場合はデータを出力します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定しても、Illegal function call のエラーは発生しません。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。(詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。)
- 「外部に出力する」が OFF の場合でも、マクロ関数を使用したデータ出力処理は実行されます。

・使用例

通信コマンドマクロで、「外部に出力する」を ON にします。

```
SetMeasureOut 1
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「GetMeasureOut (P.8-125)」
「SendData (P.8-246)」
「SendString (P.8-248)」

SetPlcData

WritePlcMemory 関数で書き込むデータを作成します。

・書式

SetPlcData <ioIdent>, <writeData()>, <offset>, <size>, <data>

・パラメータ

パラメータ名	データ型	説明
<ioIdent>	文字列型	使用する通信モジュールの識別名 （参照：「9-4 IO モジュール一覧 (P.9-25)」）
<writeData()>	整数型配列	書き込むデータ
<offset>	整数型	書き込むデータの先頭からデータを設定するアドレスまでのオフセット（バイト単位）
<size>	整数型	設定するデータのサイズ（バイト単位）
<data>	整数型 実数型 文字列型	設定するデータ

・戻り値

なし

・解説

<ioIdent> パラメータで指定した通信モジュールを使用して、<writeData()> パラメータで指定したデータ配列の先頭から<offset> パラメータで指定した値だけオフセットした位置に、<size> パラメータで指定したサイズ分の<data> パラメータで指定したデータを設定します。
このマクロ関数でデータを作成した後、WritePlcMemory 関数を実行して PLC のメモリ領域に値を書き込んでください。
<writeData()> パラメータには、書き込むデータを格納する 1 次元の整数型配列変数を、要素番号を指定せず() だけ付加して指定してください。
<offset> パラメータと<size> パラメータには、バイト単位のオフセットとサイズを指定してください。WritePlcMemory 関数で指定する単位（チャネル単位）とは異なります。
<size> パラメータには、2／4／8 のいずれかを指定してください。それぞれ、2 バイト整数／4 バイト整数／8 バイト実数として設定されます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- PLC リンク通信で WritePlcMemory 関数を使用して PLC のメモリ領域に値を書き込む場合は、必ずこのマクロ関数を使用して書き込むデータを作成してください。このマクロ関数を使用せずに WritePlcMemory 関数のパラメータに直接値を設定する場合は、正しい値を設定できない可能性があります。

・使用例

通信コマンドマクロで、PLC リンクで接続された PLC に処理ユニット番号 2 のサーチ処理ユニットの計測座標 X と計測座標 Y を書き込みます。計測座標 X は外部参照データ番号 6、計測座標 Y は外部参照データ番号 7 です。

```
IOMODULE$ = "UdpPlcLink"

Rem 計測結果を取得します
GetUnitData 2, 6, X#
GetUnitData 2, 7, Y#

Rem 実数の計測値を 1000 して整数値に変換します
VALUE0& = Int(X# * 1000)
VALUE1& = Int(Y# * 1000)

Rem データ出力エリアの設定を取得します
GetSystemData IOMODULE$, "outputArea", AREA&
GetSystemData IOMODULE$, "outputMemoryAddress", ADDRESS&

Rem 書き込むデータを整数型配列変数に格納します
Dim DATA&(1)
SetPlcData IOMODULE$, DATA&(), 0, 4, VALUE0&
SetPlcData IOMODULE$, DATA&(), 4, 4, VALUE1&

Rem データ出力エリアにデータ（4CH 分）を書き込みます
WritePlcMemory IOMODULE$, AREA&, ADDRESS&, 4, DATA&()
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 4.20 以降

・関連項目

「GetPlcData (P.8-126)」
「ReadPlcMemory (P.8-210)」
「WritePlcMemory (P.8-329)」

SetPollingState

通信モジュールの実行状態を設定します。

・書式

SetPollingState <ioldent>, <state>

・パラメータ

パラメータ名	データ型	説明
<ioldent>	文字列型	実行状態を設定する通信モジュールの識別名（参照：「9-4 IO モジュール一覧 (P.9-25)」）
<state>	整数型	設定する通信モジュールの実行状態 False：停止状態 True：動作状態

・戻り値

なし

・解説

<ioldent> パラメータで指定した通信モジュールに、<state> パラメータで指定した実行状態を設定します。

パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。

パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。

マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数で通信モジュールの実行状態を停止状態にしたあとは、必ず実行状態を動作状態に設定しなおしてください。通信モジュールの実行状態が停止状態のままの場合は、通信コマンドを受信できません。

・使用例

ユニットマクロ処理ユニットの*MEASUREPROC サブルーチンで、無手順 TCP 通信のデータを受信します。

```
Rem データを受信するために 12 バイト受信できるバッファを用意します
Dim BUFFER&(11)
IOMODULE$ = "TcpNormal"

Rem データを受信するために通信モジュールの実行状態を停止状態に設定します
SetPollingState IOMODULE$, False

Rem 受信データサイズを初期化します
SIZE& = 0

Rem データを受信するまで受信処理を繰り返します
Try
   Do
      Rem データの受信を試みます
      ReceiveData IOMODULE$, BUFFER&(), 12, SIZE&

      Rem データを受信したらデータサイズをシステム状態監視コンソールウィンドウに表示します
      If(SIZE& > 0) Then
         Print "Received data size = " + Str$(SIZE&)
      Endif
   Loop While SIZE& = 0

   Rem データを受信したので通信モジュールの実行状態を動作状態に設定します
   SetPollingState IOMODULE$, True

Catch
   Rem 停止状態にしている通信モジュールの実行状態を動作状態に戻します
   If GetPollingState(IOMODULE$) = False Then
      SetPollingState IOMODULE$, True
   Endif

End Try
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 4.20 以降

・関連項目

「GetPollingState (P.8-127)」
「ReceiveData (P.8-212)」
「SendData (P.8-246)」
「SendString (P.8-248)」

SetProfileData

設定ファイル（.ini ファイル）のデータを設定します。

・書式

SetProfileData <fileName>, <section>, <key>, <data>

・パラメータ

パラメータ名	データ型	説明
<fileName>	文字列型	対象ファイルの絶対パス名
<section>	文字列型	対象のセクション名
<key>	文字列型	対象のキー名
<data>	文字列型	設定するデータ

・戻り値

なし

・解説

<fileName>パラメータで指定したファイルの、<section>パラメータで指定したセクション名と<key>パラメータで指定したキー名に対して、<data>パラメータに指定した値を.ini ファイルの書式で書き込みます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

.ini ファイルの Setting セクション内のキー名 Logging に、フォルダパスのデータを設定します。

```
SetProfileData "E:¥Setting.ini", "Setting", "Logging", "E:¥Logging_Folder"
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 6.40 以降

・関連項目

「DeleteProfileData (P.8-65)」
「GetProfileData (P.8-130)」

SetSceneData

シーン制御マクロに関するデータやシーン変数の値を設定します。

・書式

SetSceneData <dataIdent>, <data>

・パラメータ

パラメータ名	データ型	説明
<dataIdent>	文字列型	設定するデータの識別名
<data>	整数型 実数型 文字列型	設定するデータ

・戻り値

なし

・解説

<dataIdent> パラメータで指定した識別名に対応する処理を実行します。
<dataIdent> パラメータには、シーン制御マクロのプログラムで使用している変数名を指定できるほか、次の識別名を指定できます。

- シーン変数名：<data> で指定された値を、「シーン変数名」の変数に設定します。シーン変数名は、TDM エディタのシーン変数登録画面で表示されている"SC." 以降の文字列です。
- "direct"：<data> で指定されたシーン制御マクロの処理を実行します。
- "gosub"：<data> で指定されたシーン制御マクロのサブルーチンを実行します。

<dataIdent> パラメータに"direct" や"gosub" を指定して処理を実行中にエラーが発生した場合は、シーン制御マクロのエラーとして、該当箇所でエラーが発生します。その後、このマクロ関数を呼び出したモジュールのエラーとして、このマクロ関数で Illegal function call のエラーが発生します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定しても、Illegal function call のエラーは発生しません。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）

・使用例

シーン変数に任意の値を設定します。

```
Rem シーン変数に SC.ABC& という変数を登録している場合に整数値 123 を設定します
SetSceneData "ABC&", 123

Rem シーン変数に SC.ASDF&() という配列変数を登録している場合に、
Rem ひとつ目の配列要素に整数値 123 を設定します
SetSceneData "ASDF&(0)", 123
```

通信コマンドマクロで、受信した通信コマンドのパラメータをシーン制御マクロの変数に設定して、適切なユニットに値を適用するサブルーチンを実行します。

```
Rem 通信コマンドで受信したパラメータをシーン制御マクロの変数に設定します
SetSceneData "PARAM0&", ARGUMENTVALUE#(0)
SetSceneData "PARAM1&", ARGUMENTVALUE#(1)

Rem あらかじめシーン制御マクロで定義しておいたUPDATE_PARAM サブルーチンを実行します
SetSceneData "gosub", "*UPDATE_PARAM"

Rem あらかじめシーン制御マクロで設定しておいたPrint コマンド処理を実行します
SetSceneData "direct", "print ValueN&"
```

・使用可能なモジュール

- シーン変数に対して適用する場合
  ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ
- シーン変数以外に対して適用する場合
  通信コマンドマクロ

・対応バージョン

バージョン 5.20 以降
（シーン変数に対しての適用は、バージョン 6.30 以降）

・関連項目

「GetSceneData (P.8-131)」

SetSceneDescription

シーンの説明を設定します。

・書式

SetSceneDescription <sceneNo>, <sceneDescription>

・パラメータ

パラメータ名	データ型	説明
<sceneNo>	整数型	説明を取得するシーンのシーン番号（0～127）
<sceneDescription>	文字列型	シーンの説明

・戻り値

なし

・解説

<sceneNo> パラメータで指定したシーン番号のシーンの説明に、<sceneDescription> パラメータで指定した説明を設定します。
シーンの説明は、SetSceneDescription 関数を実行するか、シーンメンテナンス画面で設定することができます。（詳細は、『画像処理システム FH/FHV シリーズ ユーザーズマニュアル (SDNB-719)』の「シーンを編集する」を参照してください。）
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）

・使用例

シーン 1 の説明を取得して、説明が未設定の場合には説明を設定します。

```
Rem シーンの説明を取得します
DESCRIPTION$ = SceneDescription$(1)

If DESCRIPTION$ = "" Then
   Rem シーンの説明を設定します
   SetSceneDescription 1, "Description 1"
Endif
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「SceneMaker$ (P.8-240)」
「SceneDescription$ (P.8-236)」
「SetSceneTitle (P.8-269)」
「SceneTitle$ (P.8-242)」
「SetSceneMaker (P.8-268)」

SetSceneGroupData

指定した識別名のシーングループデータの内容を設定します。

・書式

SetSceneGroupData <sceneGroupNo>, <dataIdent>, <data>

・パラメータ

パラメータ名	データ型	説明
<sceneGroupNo>	整数型	シーングループ番号
<dataIdent>	文字列型	シーングループデータの識別名 <dataIdent> パラメータには次の識別名を指定できます。 "saveReload" ：<sceneGroupNo> には-1, <data> には 1 を指定します。現在のシーングループの再読み込みを実行します。シーングループの再読み込みを実行すると現在のシーングループは保存されます。 "sceneGroupTitle" : <data> で指定された文字列をシーングループタイトルに設定します。シーングループタイトルは、言語識別子を使った多言語書式で指定できます。 言語識別子を使った記載の方法については、『画像処理システム FH/FHV シリーズ ユーザーズマニュアル (SDNB-719)』の「文字を入力する」を参照してください。
<data>	文字列型	設定するデータ

・戻り値

なし

・解説

<sceneGroupNo> パラメータで指定したシーングループに対して、<dataIdent> パラメータで指定した識別名に対応する処理を実行します。

パラメータとして、存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合でも、エラーは発生しません。

マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。(詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。)

・使用例

現在のシーングループのタイトルを変更します。

```
MeasureStop
  SetSceneGroupData -1, "sceneGroupTitle", "¥jpn¥サンプル¥eng¥Sample"
MeasureStart
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン

バージョン 6.50 以降

・関連項目

「GetSceneGroupData (P.8-132)」
「MeasureStart (P.8-182)」
「MeasureStop (P.8-183)」

SetSceneGroupTitle

シーングループのタイトル名を設定します。

・書式

SetSceneGroupTitle <sceneGroupNo>, <title>

・パラメータ

パラメータ名	データ型	説明
<sceneGroupNo>	整数型	タイトル名を設定するシーングループのシーングループ番号（-1 のみ）
<title>	文字列型	シーングループのタイトル名

・戻り値

なし

・解説

<sceneGroupNo> パラメータで指定したシーングループ番号のシーングループのタイトル名に、<title> パラメータで指定したタイトル名を設定します。
<title> パラメータに 32 文字以上のタイトル名を指定した場合は、先頭の 31 文字がタイトル名に設定されます。
シーングループのタイトル名は、SetSceneGroupTitle 関数を実行するか、シーンメンテナンス画面で設定することができます。（詳細は、『画像処理システム FH/FHV シリーズ ユーザーズマニュアル (SDNB-719)』の「シーンを編集する」を参照してください。）
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）

・使用例

現在のシーングループのタイトル名を設定します。

```
SetSceneGroupTitle -1, "Title"
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「SceneGroupTitle$ (P.8-239)」
「SceneTitle$ (P.8-242)」
「SetSceneTitle (P.8-269)」

SetSceneMaker

シーンの作成者名を設定します。

・書式

SetSceneMaker <sceneNo>, <sceneMaker>

・パラメータ

パラメータ名	データ型	説明
<sceneNo>	整数型	作成者名を設定するシーンのシーン番号（0～127）
<sceneMaker>	文字列型	シーンの作成者名

・戻り値

なし

・解説

<sceneNo> パラメータで指定したシーン番号のシーンの作成者に、<sceneMaker> パラメータで指定した作成者名を設定します。
<sceneMaker> パラメータに 32 文字以上の作成者名を指定した場合は、先頭の 31 文字が作成者名に設定されます。
シーンの作成者名は、SetSceneMaker 関数を実行するか、シーンメンテナンス画面で設定することができます。（詳細は、『画像処理システム FH/FHV シリーズ ユーザーズマニュアル (SDNB-719)』の「シーンを編集する」を参照してください。）
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）

・使用例

シーン 3 の作成者名を取得して、未設定であれば作成者名を設定します。

```
Rem シーンの作成者名を取得します
NAME$ = SceneMaker$(3)

If NAME$ = "" Then
   Rem シーンの作成者名を設定します
   SetSceneMaker 3, "Maker"
Endif
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「SceneDescription$ (P.8-236)」
「SceneMaker$ (P.8-240)」
「SceneTitle$ (P.8-242)」
「SetSceneDescription (P.8-265)」

「SetSceneTitle (P.8-269)」

SetSceneTitle

シーンのタイトル名を設定します。

・書式

SetSceneTitle <sceneNo>, <title>

・パラメータ

パラメータ名	データ型	説明
<sceneNo>	整数型	タイトル名を設定するシーンのシーン番号（0～127）
<title>	文字列型	シーンのタイトル名

・戻り値

なし

・解説

<sceneNo> パラメータで指定したシーン番号のシーンのタイトル名に、<title> パラメータで指定したタイトル名を設定します。
<title> パラメータに 32 文字以上のタイトル名を指定した場合は、先頭の 31 文字がタイトル名に設定されます。
シーンのタイトル名は、SetSceneTitle 関数を実行するか、シーンメンテナンス画面またはフロー編集画面で設定することができます。（詳細は、『画像処理システム FH/FHV シリーズ ユーザーズマニュアル (SDNB-719)』の「シーンを編集する」、「シーン内の処理ユニットを編集する」を参照してください。）
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）

・使用例

シーン 2 のタイトル名を設定します。

```
SetSceneTitle 2, "Title"
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「SceneDescription$ (P.8-236)」
「SceneTitle$ (P.8-242)」
「SetSceneMaker (P.8-268)」
「SceneMaker$ (P.8-240)」
「SetSceneDescription (P.8-265)」

SetStop

プログラム実行中断の条件を設定します。

・書式

SetStop <string>

・パラメータ

パラメータ名	データ型	説明
<string>	文字列型	実行中断条件のラベル

・戻り値

なし

・解説

<string> パラメータで指定した文字列を、Stop 関数によるプログラム実行を中断する条件として設定します。
SetStop 関数で設定した実行中断条件の文字列を Stop 関数のパラメータとして指定することで、実行形態がデバッグモードのときにプログラムの実行を中断できます。<string> パラメータには、ワイルドカードとして*（文字列パターン）と?（1 文字）を使用することができます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

デバッグモード時に、SetStop 関数で指定した条件でプログラムの実行を中断します。

```
Rem 実行形態をデバッグモードに設定します
Debug 18
SetStop "AA?"
SetStop "B*"

Rem 「AA?」と一致する文字列「AAB」を Stop 関数で指定した場合はプログラムが中断します
Stop "AAB"

Rem 「AA?」と一致しない文字列「AABB」を Stop 関数で指定した場合はプログラムは中断されません
Stop "AABB"

Rem 「B*」と一致する文字列「BCDEF」を Stop 関数で指定した場合はプログラムが中断します
Stop "BCDEF"

Rem 「B*」と一致しない文字列「CDEF」を Stop 関数で指定した場合はプログラムは中断されません
Stop "CDEF"

Rem 実行形態をリリースモードに設定します
Debug 1
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 5.20 以降

・関連項目

「Cls (P.8-48)」
「Cont (P.8-49)」
「Debug (P.8-63)」
「DebugPrint (P.8-64)」
「List (P.8-167)」
「Print (P.8-202)」
「SetVar (P.8-281)」
「Stop (P.8-286)」
「SubList (P.8-292)」
「VarList (P.8-322)」

SetSystemData

システムデータまたは、システム変数に値を設定します。

・書式

SetSystemData <dataIdent0>, <dataIdent1>, <data>

・パラメータ

パラメータ名	データ型	説明
<dataIdent0>	文字列型	設定するシステムデータの識別情報 0 のデータ識別名
<dataIdent1>	文字列型	設定するシステムデータの識別情報 1 のデータ識別名
<data>	整数型 実数型 文字列型	設定するシステムデータの値

・戻り値

なし

・解説

- システムデータに値を設定する場合
  <dataIdent0> パラメータで指定した識別情報 0 に属する、<dataIdent1> パラメータで指定した識別情報 1 のシステムデータに<data> パラメータで指定した値を設定します。システムデータの識別情報については、「9-3 システムデータ一覧 (P.9-10)」をご覧ください。
- システム変数に値を設定する場合
  <dataIdent0> に"macro_sy" という文字列を設定し、<dataIdent1> に TDM エディタのシステム変数登録画面で表示されている"SY." 以降の変数名を設定します。

パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない識別名を指定した場合は、Illegal function call のエラーが発生します。
<dataIdent1> パラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。(詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。)

・使用例

システム変数に任意の値を設定します。

```
Rem システム変数に SY.ABC& という変数を登録している場合に整数値 123 を設定します
SetSystemData "macro_sy", "ABC&", 123

Rem システム変数に SY.ASDF&() という配列変数を登録している場合に、
Rem ひとつめの配列要素に整数値 123 を設定します
SetSystemData "macro_sy", "ASDF&(0)", 123
```

識別情報 0 が"Measure" の計測制御設定に属する、識別情報 1 が"captureDirectory" の画面キャプチャ保存先フォルダに、"E:¥temp¥bmp" の値を設定します。

```
Rem 画面キャプチャ保存先フォルダとして設定する保存先のパスを求めます
DIRNAME$ = "E:¥temp¥bmp"

Rem 計測制御設定に属する画面キャプチャ保存先フォルダを設定します
SetSystemData "Measure", "captureDirectory", DIRNAME$
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降
（システム変数に対しての適用はバージョン 6.30 以降）

・関連項目

「AddSystemData (P.8-24)」
「GetSystemData (P.8-133)」
「SetGlobalData (P.8-253)」
「SetUnitData (P.8-275)」

SetTextStyle

文字列の描画属性を設定します。

・書式

SetTextStyle <fontSize>, <align>, <color>, <angle>, <style>

・パラメータ

パラメータ名	データ型	説明
<fontSize>	整数型	描画する文字列のフォントサイズ
<align>	整数型	描画する文字列の配置 TA_BASELINE：ベースライン揃え TA_BOTTOM：下端揃え TA_TOP：上端揃え TA_CENTER：水平方向中心揃え TA_LEFT：左端揃え TA_RIGHT：右端揃え TA_NOUPDATECP：カレントポジション非更新 TA_RTLRENDING：右から左 TA_UPDATECP：カレントポジション更新
<color>	整数型	描画する文字列の色のカラー値 JUDGE_NC：未計測色（灰色） JUDGE_OK：OK 色（緑色） JUDGE_NG：NG 色（赤色） RGB 関数：任意の色
<angle>	整数型	描画する文字列の傾き（0～359）
<style>	整数型	描画する文字列の書体 FONTSTYLE_NORMAL：標準 FONTSTYLE_BOLD：太字 FONTSTYLE_ITALIC：斜体 FONTSTYLE_UNDERLINE：下線 FONTSTYLE_STRIKEOUT：取り消し線

・戻り値

なし

・解説

<fontSize> パラメータで指定したフォントサイズと、<align> パラメータで指定した文字列の配置と、<color> パラメータで指定した文字列の色と、<angle> パラメータで指定した文字列の傾きと、<style> パラメータで指定した文字列の書体を描画属性として設定します。画像表示ウィンドウ制御の DrawTextG 関数を実行する前にあらかじめこのマクロ関数を実行することで、設定した描画属性のグラフィックを描画することができます。グラフィック図形を描画するマクロ関数の描画属性を設定する場合は、SetDrawStyle 関数を使用してください。(参照：「SetDrawStyle (P.8-250)」)

<align> パラメータとして、TA_BOTTOM、TA_TOP、TA_LEFT、TA_CENTER、TA_RIGHT は OR 指定することができます。

<color> パラメータとして、RGB 関数で取得したカラー値を設定することもできます。(参照：「RGB (P.8-222)」)

パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。

整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。

マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、*MEASUREDISPI サブルーチンまたは*MEASUREDISPG サブルーチンの中でしか使用できません。それ以外のサブルーチンで使用しようとした場合は、Illegal function call のエラーが発生します。

・使用例

ユニットマクロ処理ユニットの*MEASUREDISPG サブルーチンの中で、フォントサイズが 20、配置が下揃えかつ水平方向中心揃え、色が OK 色、傾きが 90 度で斜体の文字列をグラフィック表示します。

```
*MEASUREDISPG

    Rem 描画属性を設定します
    SetTextStyle 20, TA_BOTTOM OR TA_CENTER, JUDGE_OK, 90, FONTSTYLE_ITALIC

    Rem グラフィックを描画します
    DrawTextG "Measurement OK", 100, 100, 0, UnitNo

Return
```

・使用可能なモジュール

ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「DrawTextG (P.8-99)」
「RGB (P.8-222)」
「SetDrawStyle (P.8-250)」

SetTextWindow

テキスト表示ウィンドウの状態を設定します。

・書式

SetTextWindow <unitNo>, <subNo>, <update>, <visible>

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	表示対象の処理ユニットの処理ユニット番号
<subNo>	整数型	表示対象のサブ番号
<update>	整数型	更新タイミング（常に 0）
<visible>	整数型	表示有無（このパラメータは常に無効です）

・戻り値

なし

・解説

テキスト表示ウィンドウの状態を設定します。
<unitNo> パラメータには、表示対象の処理ユニット番号の値を指定してください。テキスト表示ウィンドウの表示対象をフロー表示に連動させる場合は、-1 を設定してください。
<subNo> パラメータには、表示対象のサブ画像番号の値として通常は 0 を指定してください。ユニットマクロで使用する場合は、MEASUREDISPT サブルーチンの中で DisplaySubNo マクロ関数と組み合わせて実行することで、0 以外のサブ画像番号のテキスト表示ウィンドウの状態を設定することができます。
<update> パラメータには、更新タイミングの値として常に 0 を指定してください。
<visible> パラメータには、値を設定しても設定内容は反映されません。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）

・使用例

通信コマンドマクロで、テキスト表示ウィンドウで表示する処理ユニットの処理ユニット番号を、通信コマンドの引数で指定した番号に変更します。

```
Rem テキスト表示ウィンドウの状態を取得します
GetTextWindow UNITNO&, SUBNO&, UPDATE&, VISIBLE&

Rem 表示する処理ユニット番号にコマンドの引数で指定された番号を設定します
SetTextWindow argumentValue#(0), SUBNO&, UPDATE&, VISIBLE&
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「DisplaySubNo (P.8-69)」
「DisplayUnitNo (P.8-70)」
「GetTextWindow (P.8-137)」
「RefreshTextWindow (P.8-217)」
「SetDisplayUnitNo (P.8-249)」
「SetImageWindow (P.8-254)」

「UnitNo (P.8-318)」　　「Ut (P.8-320)」

SetUnitData

処理ユニットのデータを設定します。

・書式

SetUnitData <unitNo>, <dataNo>, <data>
SetUnitData <unitNo>, <dataIdent>, <data>

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	処理ユニット番号（0～（現在シーンの処理ユニット数 -1)）
<dataNo>	整数型	取得する処理ユニットデータの外部参照データ番号（参照：『画像処理システム FH/FHV シリーズ 処理項目機能リファレンスマニュアル（SDNB-713)』）
<dataIdent>	文字列型	取得する処理ユニットデータのデータ識別名
<data>	整数型 実数型 文字列型	取得した処理ユニットデータ

・戻り値

なし

・解説

<unitNo> パラメータで指定した処理ユニットの持つ、<dataNo> パラメータで指定した外部参照データ番号のデータに、<data> パラメータで指定したデータを設定します。<dataNo> パラメータの代わりに<dataIdent> パラメータを指定してデータを取得することもできます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。(詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。)
- このマクロ関数は、1 回以上計測するか画像ファイルを指定して再計測した後、計測画像が表示された状態で使用してください。
- このマクロ関数は、数値で設定すべき箇所に文字列で設定すると数値に自動的に変換されます。そのため処理ユニットの数値データに設定を行う場合、数値を含む文字列を設定すると、先頭から数値として有効な範囲を認識し設定します。

・使用例

処理ユニット番号 2 のサーチ処理ユニットの「総合判定反映」を「なし」に設定します。「総合判定反映」は外部参照データ番号 103、外部参照データ識別名"overallJudge" です。

```
SetUnitData 2, 103, 1

Rem 103 の代わりに"overallJudge" を指定しても同じ結果が得られます
SetUnitData 2, "overallJudge", 1
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「GetUnitData (P.8-138)」
「GetUnitFigure (P.8-139)」
「MeasureStart (P.8-182)」
「MeasureStop (P.8-183)」
「UnitNo (P.8-318)」
「Ut (P.8-320)」

SetUnitFigure

処理ユニットの図形データを設定します。

・書式
SetUnitFigure <unitNo>, <figureNo>, <figure()>

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	処理ユニット番号（0～（現在シーンの処理ユニット数 -1)）
<figureNo>	整数型	設定する図形番号（参照：「9-7 図形番号一覧 (P.9-70)」)
<figure()>	整数型配列	設定する図形データ（参照：「9-7 図形番号一覧 (P.9-70)」

・戻り値
なし

・解説
<unitNo> パラメータで指定した処理ユニットの、<figureNo> パラメータで指定した図形に、<figure()> パラメータで指定した図形データを設定します。図形データにモデル図形を設定した場合、モデルを再登録します。
<figure()> パラメータには、図形データを格納する 1 次元の整数型配列変数を、要素番号を指定せず () だけ付加して指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。(詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。)
- このマクロ関数は、1 回以上計測するか画像ファイルを指定して再計測した後、計測画像が表示された状態で使用してください。
- 設定する図形に計測画像外の画素が含まれないように図形データを設定してください。

・使用例
処理ユニット番号 2 のサーチ処理ユニットの領域図形（長方形）の位置を変更します。

```
Dim FIGURE&(5)

Rem 処理ユニットの図形データを取得します
GetUnitFigure 2, 1, FIGURE&()

Rem 取得した図形データを元に、領域図形の左上点 XY 座標と右下点 XY 座標の値を変更します
FIGURE&(2) = 100
FIGURE&(3) = 100
FIGURE&(4) = 300
FIGURE&(5) = 300

Rem 処理ユニットに領域図形の位置を変更した図形データを設定します
SetUnitFigure 2, 1, FIGURE&()
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「CopyUnitFigure (P.8-55)」
「FigureType (P.8-116)」
「GetUnitData (P.8-138)」
「GetUnitFigure (P.8-139)」
「MeasureStart (P.8-182)」
「MeasureStop (P.8-183)」
「SetUnitData (P.8-275)」
「UnitNo (P.8-318)」
「Ut (P.8-320)」

SetUnitJudge

処理ユニットの判定結果を設定します。

・書式

SetUnitJudge <unitNo>, <judge>[, <totalJudgeRefrect>]

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	処理ユニット番号（0～（現在シーンの処理ユニット数 -1））
<judge>	整数型	設定する判定結果 JUDGE_NC：判定なし（未計測） JUDGE_OK：判定結果 OK JUDGE_NG：判定結果 NG JUDGE_IMAGEERROR：判定結果エラー（画像形式不整合） JUDGE_MODELERROR：判定結果エラー（モデル未登録） JUDGE_MEMORYERROR：判定結果エラー（メモリ不足） JUDGE_ERROR：判定結果エラー（その他エラー）
<totalJudgeRefrect>	整数型	総合判定反映有無 False：反映しない True：反映する

・戻り値

なし

・解説

<unitNo> パラメータで指定した処理ユニットの判定結果に、<judge> パラメータで指定した判定結果を設定します。<totalJudgeRefrect> パラメータを省略した場合は、設定した判定が総合判定にも反映されます。

このマクロ関数で他の処理ユニットの判定結果を設定する場合は、以下のような動作になります。
- 1 回の計測でこのマクロ関数により判定結果を設定したあと、処理ユニットの計測処理を実行した場合、このマクロ関数で設定した判定結果は処理ユニットの計測処理の結果で上書きされます。
- 1 回の計測で処理ユニットの計測処理を実行したあと、このマクロ関数で判定結果を設定した場合、処理ユニットの判定結果はこのマクロ関数で設定した判定結果で上書きされます。

どちらの場合も、<totalJudgeRefrect> パラメータに True を指定することで、上書きされる前の判定結果を総合判定に反映させることができます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
- ユニットマクロでは、プログラムの実行時にエラーが発生した場合、SetUnitJudge の設定にかかわらず、処理ユニットの判定結果は NG になります。
- ユニット演算マクロ自身の判定結果は、ユニット演算マクロの演算式の判定条件によって決まります。SetUnitJudge では設定できません。

・使用例
処理ユニット番号 2 のサーチ処理ユニットの判定結果を判定結果 OK に設定します。

```
SetUnitJudge 2, JUDGE_OK, True
```

・使用可能なモジュール
ユニット演算マクロ ／ ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「SetUnitData (P.8-275)」
「TotalJudge (P.8-300)」
「UnitJudge (P.8-317)」
「UnitNo (P.8-318)」
「Ut (P.8-320)」

SetUnitTitle

処理ユニットのタイトル名を設定します。

・書式
SetUnitTitle <unitNo>, <title>

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	処理ユニット番号（0～（現在シーンの処理ユニット数 -1））
<title>	文字列型	設定する処理ユニットのタイトル名（最大 31 文字）

・戻り値
なし

・解説
<unitNo> パラメータで指定した処理ユニットのタイトル名に、<title> パラメータで指定したタイトルを設定します。<title> パラメータに 32 文字以上のタイトル名を指定した場合は、先頭の 31 文字がタイトル名に設定されます。

パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）

・使用例

処理ユニット番号 2 のサーチ処理ユニットのタイトル名にサーチ対象名として"bolt search" を設定します。

```
SetUnitTitle 2, "bolt search"
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「MeasureStart (P.8-182)」
「MeasureStop (P.8-183)」
「UnitNo (P.8-318)」
「UnitTitle$ (P.8-319)」
「Ut (P.8-320)」

SetUserSubroutine

外部 DLL に定義されたユーザ定義関数を登録します。

・書式

SetUserSubroutine <subroutineIdent>, <dllFileName>, <functionName>

・パラメータ

パラメータ名	データ型	説明
<subroutineIdent>	文字列型	登録するユーザ定義関数の識別名
<dllFileName>	文字列型	DLL ファイル名
<functionName>	文字列型	登録するユーザ定義関数名

・戻り値

なし

・解説

<dllFileName> パラメータで指定した DLL ファイル内で定義されている<functionName> パラメータで指定した関数名の関数を、<subroutineIdent> パラメータで指定した識別名で登録します。この関数で登録したユーザ定義関数は、識別名を指定することで Call 関数を使って呼び出せるようになります。
<dllFileName> パラメータには、DLL ファイル名の拡張子（.dll）を除いたファイル名を指定してください。

この関数は、通常は*MCRINIT サブルーチンの中または Call 関数を実行するまでに実行してください。
ユーザ定義関数の登録は、SetUserSubroutine 関数を実行したマクロカスタマイズ機能の中でのみ有効です。ユーザ定義関数を呼び出して実行する処理ユニットごと、各シーンのシーン制御マクロごと、通信コマンドマクロのコマンドごとに実行してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
<subroutineIdent> パラメータまたは<functionName> パラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
<dllFileName> パラメータとして、絶対パスが 255 文字を超えるファイルのファイル名文字列を指定した場合は、String too long のエラーが発生します。
この関数で登録したユーザ定義関数がマクロ関数として正しいインタフェースで実装されていなかった場合でもエラーは発生しません。この場合 Call 関数でユーザ定義関数の処理を実行した際にエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数では、FZ-CoreRA.exe と同じフォルダに存在する DLL に定義された、正しいインタフェースでプログラミングされたユーザ定義関数しか使用できません。ユーザ定義関数の作成には FH-AP1 をご利用ください。

・使用例

MacroUserProc.dll に定義されているユーザ定義関数"UserProc0" を、識別名"USR" として登録します。その後登録したユーザ定義関数を、識別名を指定して呼び出して実行します。

```
Rem ユーザ定義関数をプログラムの中で使えるように登録します
SetUserSubroutine "USR", "MacroUserProc", "UserProc0"

Rem 登録したユーザ定義関数を呼び出して実行します
Call "USR", 0
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 5.20 以降

・関連項目

「Call（ユーザ定義関数実行）（P.8-36)」

SetVar

指定した変数名の変数を一括で設定します。

・書式

SetVar <variableName>, <value>

・パラメータ

パラメータ名	データ型	説明
<variableName>	文字列型	設定対象の変数の変数名
<value>	整数型 実数型 文字列型	設定する値

・戻り値

なし

・解説

<variableName> パラメータで指定した変数に、<value> パラメータで指定した値を設定します。
<value> パラメータには、<variableName> パラメータで指定した変数と同じデータ型の値を設定してください。
<variableName> パラメータには、ワイルドカードとして*（文字列パターン）と?（1 文字）を使用することができます。<variableName> パラメータでワイルドカードを使用して複数の変数を指定する場合は、指定する変数が同じデータ型であることを確認してください。
ワイルドカードでは次のような指定が可能です。

*	すべての変数を指定します
???	3 文字の変数名の変数を指定します
A*	A で始まる変数名の変数を指定します
*A*A*A*	A を 3 文字以上含む変数名の変数を指定します
????*	4 文字以上の変数名の変数を指定します

パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定しても、Illegal function call のエラーは発生しません。
<variableName> パラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）

・使用例

処理ユニット番号 1 のユニットマクロ処理ユニットの参照変数として登録した A01@、A02@、AA01@、AB01@、B01@、B02@ のうち、頭文字が「A」の参照変数 A01@、A02@、AA01@、AB01@ の値を一括して設定します。

```
A01@ = 100
A02@ = 100
AA01@ = 100
AB01@ = 100
B01@ = 100
B02@ = 100

Rem 変数名の先頭文字が A の変数にだけ 123 を設定します
SetVar "A*@", 123
```

結果は以下のとおりになります。

```
A01@ = 123
A02@ = 123
AA01@ = 123
AB01@ = 123
B01@ = 100
B02@ = 100
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 5.20 以降

・関連項目

「Cont (P.8-49)」
「Debug (P.8-63)」
「DebugPrint (P.8-64)」
「List (P.8-167)」
「Print (P.8-202)」
「SetStop (P.8-270)」
「Stop (P.8-286)」
「VarList (P.8-322)」

Sin

指定した数式の正弦（サイン）を求めます。

・書式

Sin(<expression>)

・パラメータ

パラメータ名	データ型	説明
<expression>	整数型 実数型	正弦を求める数式

・戻り値

実数型の正弦の値を-1 から 1 までの範囲の値で返します。

・解説

<expression> パラメータで指定した数式の正弦を求めます。
<expression> パラメータには値をラジアン値で指定します。角度値をラジアン値に変換するには、角度値に π/180 をかけます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

30° の正弦を求めます。

```
DATA# = Sin(30/180*3.141592)
```

結果は以下のとおりになります。

```
DATA# = 0.5
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「Atn (P.8-33)」
「Cos (P.8-59)」
「GetUnitData (P.8-138)」
「Tan (P.8-295)」
「UnitData (P.8-311)」

Sqr

平方根を求めます。

・書式

Sqr(<expression>)

・パラメータ

パラメータ名	データ型	説明
<expression>	整数型 実数型	平方根を求める数式

・戻り値

実数型の平方根の値を返します。

・解説

<expression> パラメータで指定した数式の平方根を求めます。
<expression> パラメータには、0 または正の数を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

256 の平方根を求めます。

```
DATA# = Sqr(256)
```

結果は以下のとおりになります。

```
DATA# = 16
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「GetUnitData (P.8-138)」　「UnitData (P.8-311)」

StartTimer

経過時間の測定を開始します。

・書式
StartTimer

・パラメータ
なし

・戻り値
Timer 関数の<start>パラメータに指定するための、実数型の値を返します。

・解説
経過時間の測定を開始します。
このマクロ関数で経過時間の測定を開始した後、Timer 関数を実行して経過時間を取得してください。
ElapsedTime 関数がユニット演算マクロやユニットマクロなどの処理ユニットでしか実行できないのに対して、このマクロ関数と Timer 関数はすべてのマクロカスタマイズ機能で実行できます。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
測定する対象となる処理の処理時間を 4 回測定し、表示します。

```
Dim TIME&(3)


For NUM& = 0 To 3
T# = StartTimer
Rem 測定する対象となる処理を実行します

Rem StartTimer 関数の戻り値を使って経過時間を取得します
TIME&(NUM&) = Timer(T#, 0)
Next

For NUM& = 0 To 3
Print(TIME&(NUM&))
Next
```

StartTimer 関数と Timer 関数の間に、Print 関数などの測定する対象以外の処理を入れないでください。

・使用可能なモジュール
ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「ElapsedTime (P.8-103)」
「Timer (P.8-298)」
「Wait (P.8-328)」

Stop

プログラムの実行を中断します。

・書式

Stop [<string>]

・パラメータ

パラメータ名	データ型	説明
<string>	文字列型	実行中断条件のラベル このパラメータは省略可能です。 パラメータを指定しない場合は、本関数が実行された時点でプログラムの実行が中断されます。

・戻り値

なし

・解説

<string> パラメータで指定した文字列が、あらかじめ SetStop 関数を実行して設定されたプログラムの実行中断条件の文字列と一致した場合に、プログラムの実行を中断します。
中断したプログラムを再開するにはシステム状態監視コンソールウィンドウから Cont 関数を使用してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、Debug 関数でデバッグモードを指定している場合にのみ有効です。それ以外の場合は、Rem 関数と同様に、プログラム行の本関数の記述は無視されます。(詳細は、「6-1 デバッグ機能の使い方について (P.6-2)」を参照してください。
- システム状態監視コンソールウィンドウにデータを出力すると、システム状態監視コンソールウィンドウがセンサコントローラのメイン画面の前面に表示されます。メイン画面を前面に表示する場合は、システム状態監視コンソールウィンドウ右上の[_]をクリックするか、センサコントローラに USB キーボードを接続して、[Alt] + [Tab] で表示する画面を切り替えてください。

・使用例

デバッグモード時に、SetStop 関数で指定した条件でプログラムの実行を中断します。

```
Rem 実行形態をデバッグモードに設定します
Debug 18
SetStop "ABC"

Rem 「ABC」と一致する文字列「ABC」を Stop 関数で指定した場合はプログラムが中断します
Stop "ABC"

Rem 「ABC」と一致しない文字列「ABCD」を Stop 関数で指定した場合はプログラムは中断されません
Stop "ABCD"

Rem Stop 関数でパラメータを指定しない場合は本関数が実行された時点でプログラムが中断します
Stop

Rem 実行形態をリリースモードに設定します
Debug 1
```

・使用可能なモジュール
ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 5.20 以降

・関連項目

「Cont (P.8-49)」
「DebugPrint (P.8-64)」
「Print (P.8-202)」
「SetVar (P.8-281)」
「Debug (P.8-63)」
「List (P.8-167)」
「SetStop (P.8-270)」
「VarList (P.8-322)」

Str$

数値を数字文字列に変換します。

・書式
Str$(<expression>)

・パラメータ

パラメータ名	データ型	説明
<expression>	整数型 実数型	数字文字列に変換する数式

・戻り値
文字列型の数字文字列の値を返します。

・解説
<expression> パラメータで指定した数式を数字文字列に変換します。
Str$ 関数の逆関数には Val 関数があります。Val 関数は、指定した数字文字列を数値に変換します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
実数型のパラメータとして、-1.0E30～1.0E30 の範囲を超える値を指定した場合は、Overflow のエラーが発生する可能性があります。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
ユニットマクロの MEASUREPROC サブルーチンで、処理ユニット番号 5 のサーチ処理ユニットの判定結果と相関値を、システム状態監視コンソールウィンドウに出力します。判定結果は外部参照データ番号 0、相関値は外部参照データ番号 5 です。

```
*MEASUREPROC

    Rem 処理ユニットの計測結果を取得します
    GetUnitData 5, 0, JG&
    GetUnitData 5, 5, CR#

    Rem 取得した計測結果を数字文字列に変換して、Print 関数を使用してシステム状態監視コンソールウィンドウに出力します
    Print Str$(JG&) +",," + Str$(CR#)

Return
```

結果は以下のとおりになります。

```
1,98.4
```

・使用可能なモジュール
ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「Asc (P.8-31)」
「GetUnitData (P.8-138)」
「LCase$ (P.8-162)」
「Len (P.8-164)」
「Piece$ (P.8-200)」
「Right$ (P.8-223)」
「UCase$ (P.8-309)」
「Chr$ (P.8-41)」
「Hex$ (P.8-142)」
「Left$ (P.8-163)」
「Mid$ (P.8-186)」
「Print (P.8-202)」
「Str2$ (P.8-288)」
「Val (P.8-321)」

Str2$

数値を書式指定して数字文字列に変換します。

・書式
Str2$(<expression>, <integral>, <fixed>, <zeroSuppression>, <negative>)

・パラメータ

パラメータ名	データ型	説明
<expression>	整数型 実数型	数字文字列に変換する数式
<integral>	整数型	整数部の桁数（0～15）
<fixed>	整数型	小数部の桁数（0～15）
<zeroSuppression>	整数型	数値左の空き領域の埋め方 0：スペースで埋める 1：0 埋める
<negative>	整数型	負数の表現方法 0：- 1：8

・戻り値
文字列型の数字文字列の値を返します。

・解説

<expression> パラメータで指定した数式を、<integral> パラメータで指定した整数部の桁数かつ<fixed> パラメータで指定した小数部の桁数かつ<zeroSuppression> パラメータで指定した数値左の空き領域の埋め方かつ<negative> パラメータで指定した負数の表現で、数字文字列に変換します。
Str$ 関数の逆関数には Val 関数がありますが、Str2$ 関数の逆関数はありません。Val 関数は、指定した数字文字列を数値に変換します。
<integral> パラメータとして、0 を指定した場合は、<expression> パラメータで指定した数式にあるだけの整数部の桁数の数字文字列に変換します。
<integral> パラメータとして、<expression> パラメータで指定した数式にあるだけの整数部の桁数よりも小さな値を指定した場合は、指定した整数部の桁数を最大桁数（負数の場合は最小桁数）とする数値の数字文字列に変換します。

- 例）整数部の桁数 2、小数部の桁数 0 の場合
  <expression> パラメータが 179.099 のとき、変換した数字文字列は"99"
  <integral> パラメータとして、<expression> パラメータで指定した数式にあるだけの整数部の桁数よりも大きな値を指定した場合は、数式の整数部の桁数を超える桁が<zeroSuppression> パラメータで指定した空き領域の埋め方で埋められた数字文字列に変換します。
  <integral> パラメータで指定した整数部の桁数には、<negative> パラメータで指定した負数の表現が含まれます。
- 例）整数部の桁数 3、小数部の桁数 3 の場合
  <expression> パラメータが 999.999 のとき、変換した数字文字列は"999.999"
  <expression> パラメータが-999.999 のとき、変換した数字文字列は"-99.999"
  <fixed> パラメータとして、0 を指定した場合は、<expression> パラメータで指定した数式の小数点以下を四捨五入した数字文字列に変換します。
  <fixed> パラメータとして、<expression> パラメータで指定した数式にあるだけの小数部の桁数よりも小さな値を指定した場合は、<fixed> パラメータで指定した小数部の最小桁の次の桁の値を四捨五入した数値の数字文字列に変換します。
- 例）整数部の桁数 2、小数部の桁数 4 の場合
  <expression> パラメータが 10.12345 のとき、変換した数字文字列は"10.1235"
  <fixed> パラメータとして、<expression> パラメータで指定した数式にあるだけの小数部の桁数よりも大きな値を指定した場合は、数式の小数部の桁数を超える桁が 0 で埋められた数字文字列に変換します。
- 例）整数部の桁数 2、小数部の桁数 5 の場合
  <expression> パラメータが 10.123 のとき、変換した数字文字列は"10.12300"

パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
実数型のパラメータとして、-1.0E30～1.0E30 の範囲を超える値を指定した場合は、Overflow のエラーが発生する可能性があります。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
処理ユニット番号 5 のサーチ処理ユニットの相関値、計測座標 X、計測座標 Y を、システム状態監視コンソールウィンドウに出力します。相関値は外部参照データ番号 5、計測座標 X は外部参照データ番号 6、計測座標 X は外部参照データ番号 7 です。

```
Rem 処理ユニットの計測結果を取得します

                                GetUnitData 5, 6, X#
                                GetUnitData 5, 7, Y#
                                GetUnitData 5, 5, CR#

                                Rem 計測結果を書式指定して数字文字列に変換します
                                RESX$ = Str2$(X#, 3, 3, 0, 0)
                                RESY$ = Str2$(Y#, 3, 3, 0, 0)
                                RESCR$ = Str2$(CR#, 3, 0, 0, 0)

                                Rem Print 関数を使用してシステム状態監視コンソールウィンドウに出力
します
                                Print RESX$ + ",," + RESY$ + ",," + RESCR$
```

結果は以下のとおりになります。

```
150.000,359.000, 97
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「Asc (P.8-31)」
「Chr$ (P.8-41)」
「Hex$ (P.8-142)」
「LCase$ (P.8-162)」
「Left$ (P.8-163)」
「Len (P.8-164)」
「Mid$ (P.8-186)」
「Piece$ (P.8-200)」
「Print (P.8-202)」
「Right$ (P.8-223)」
「Str$ (P.8-287)」
「UCase$ (P.8-309)」
「Val (P.8-321)」

Sub ～ End Sub

サブルーチンを定義します。

・書式

Sub <label>([, <argument0>...<argumentN>])
…
End Sub

・パラメータ

パラメータ名	データ型	説明
<label>	文字列型	実行するサブルーチンのラベル名
<argument0>	整数型 実数型 文字列型 配列型 バリアント型	サブルーチンが受け取るデータ
:	:	:

パラメータ名	データ型	説明
<argumentN>	整数型 実数型 文字列型 配列型 バリアント型	サブルーチンが受け取るデータ

・戻り値
なし

・解説
マクロコード上の他のサブルーチンから実行できるサブルーチンを定義する事ができます。
<argument0> ～ <argumentN> パラメータにはすべての型のローカル変数、ローカル配列変数を指定できます。本サブルーチン内で引数として渡された <argument0> ～ <argumentN> パラメータの値を変更すると、Call 関数による呼び出し元で引数として指定された変数の値が更新されます。
ローカル配列変数を引数として使う場合は%AA&() のように配列要素番号を指定せずに配列全体を指定します。%AA&(0) のようにローカル配列変数の 1 要素を <argument> に指定する事はできません。ローカル配列変数が引数として渡された場合、ReDim コマンドによって配列サイズを変更する事ができます。
<argument0> ～ <argumentN> パラメータにローカル変数、ローカル配列変数以外を指定すると Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
ユニットマクロの*MeasureProc サブルーチンで別に定義した*ValueInit サブルーチンの処理を実行します。

```
*MCRINIT
  Dim X#(0)
  Dim Y#(0)
  Dim ANG#(0)
  Dim JG&
Return
*MEASUREPROC
  Rem 他のサブルーチンを実行します
  Call *ValueInit(X#(), Y#(), ANG#(), JG&)
Return
Rem 各種計測値を初期化する
Sub *ValueInit(%POSX#(), %POSY#(), %ANGLE#(), %JUDGE&)
  ReDim %POSX#(10)
  ReDim %POSY#(10)
  ReDim %ANGLE#(10)
  %JUDGE& = JUDGE_NC
End Sub
```

・使用可能なモジュール
ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 6.50 以降

・関連項目

「Call（構造化サブルーチン呼出）（P.8-35）」

SubList

サブルーチンの一覧をコンソールウィンドウに出力します。

・書式
SubList

・パラメータ
なし

・戻り値
なし

・解説
マクロコードで定義されている引数付きサブルーチンのリストを、システム状態監視コンソールウィンドウに出力します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
マクロコードで定義済みの引数付きサブルーチンのリストを表示します。定義されているすべてのサブルーチンの名前と引数が、システム状態監視コンソールウィンドウに出力されます。

```
SubList
Sub *TESTSUB1(%AA&, %BB#)
Sub *TESTSUB2(%AA&)
Sub *TESTSUB3(%AA&)
Ok
```

・使用可能なモジュール
ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 6.50 以降

・関連項目

「Cls (P.8-48)」
「Cont (P.8-49)」
「Debug (P.8-63)」
「DebugPrint (P.8-64)」
「List (P.8-167)」
「Print (P.8-202)」
「SetStop (P.8-270)」

SystemInfo

システムに関する情報を取得します。

・書式

SystemInfo(<kind>)

・パラメータ

パラメータ名	データ型	説明
<kind>	整数型	取得情報種別 4: Controller のモデル（0: FZ、256: FH1xx0,FH3xx0、257: FH-L550,FH-L551、258: FH2xx0,FH5xx0,FH2xx1,FH5xx1、259: FH2xx2,FH5xx2、260: FH2xx3,FH5xx3、261: FH-L552） 8: EtherCAT の有無（0: EtherCAT 通信ボード非装着、2: EtherCAT 通信ボード装着済（FH） 10: bsp のバージョン 11: fpga のバージョン 20: OS のメジャーバージョン 21: OS のマイナーバージョン 22: OS のビット数（32: 32bit、64: 64bit） 23: OS のビルドバージョン 100: カメラの最大接続数

・戻り値

取得した情報（整数値）を返します。

・解説

<kind> パラメータで指定した種別の情報を取得します。定義されていない <kind> を指定してもエラーは発生しません。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

シミュレーションソフトとして実行中かチェックします。

```
If SystemInfo(4) > 0 Then
  Rem シミュレーションソフトとして実行中のため、オフラインの再計測処理を記述
Else
  Rem カメラによる計測処理を記述
EndIf
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 6.41 以降

・関連項目

なし

SystemReset

センサコントローラを再起動します。

・書式

SystemReset

・パラメータ

なし

・戻り値

なし

・解説

センサコントローラを再起動します。
PC の場合は、何もせず終了します。

・使用上の注意

なし

・使用例

システムデータを読み込んだ後、［本体に保存］を実行します。その後読み込んだ設定をセンサコントローラに反映させるために、センサコントローラを再起動します。

```
Rem システムデータファイルを読み込みます
LoadSystemData "E:¥BACKDIR¥backupsysset.ini"

Rem 本体に保存します
SaveData

Rem センサコントローラを再起動します
SystemReset
```

・使用可能なモジュール

通信コマンドマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「Date$ (P.8-62)」
「Left$ (P.8-163)」
「SaveData (P.8-227)」
「ExitFzProcess (P.8-113)」
「LoadSystemData (P.8-171)」

Tan

指定した数式の正接（タンジェント）を求めます。

・書式

Tan(<expression>)

・パラメータ

パラメータ名	データ型	説明
<expression>	整数型 実数型	正接を求める数式

・戻り値

実数型の正接の値を返します。

・解説

<expression> パラメータで指定した数式の正接を求めます。
<expression> パラメータには値をラジアン値で指定します。角度値をラジアン値に変換するには、角度値に π/180 をかけます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

45° の正接を求めます。

```
DATA# = Tan(45/180*3.141592)
```

結果は以下のとおりになります。

```
DATA# = 0.999999673205
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「Atn (P.8-33)」
「Cos (P.8-59)」
「GetUnitData (P.8-138)」
「Sin (P.8-283)」
「UnitData (P.8-311)」

TestMeasure

テスト計測を実行します。

・書式

TestMeasure [<preImageNo> ¦ <fileName>]

・パラメータ

パラメータ名	データ型	説明
<preImageNo>	整数型	テスト計測対象の画像の画像ロギング番号（-1～すでにロギングしている本体ロギング画像の枚数 -1） -1：最新入力画像 0：最新の本体ロギング画像
<fileName>	文字列型	テスト計測対象の画像の画像ファイル名

・戻り値

なし

・解説

テスト計測は、指定した画像を計測画像として計測を行うコマンドです。このコマンドを使った計測では、処理項目として計測結果を保持しない計測となります。センサコントローラの起動直後やシーン切替直後の画像入力が行われていない状態を回避するときに使用します。

<preImageNo> パラメータで指定した画像ロギング番号の画像または<fileName> パラメータで指定した画像ファイル名の画像に対して、テスト計測を実行します。

<preImageNo> パラメータには、本体ロギング画像としてすでにロギングしているロギング画像の番号を指定してください。（詳細は、『画像処理システム FH/FHV シリーズ ユーザーズマニュアル (SDNB-719)』の「ロギング条件を設定する［ロギング設定］」を参照してください。）

<preImageNo> パラメータとして、-1 を指定した場合は、最新入力画像に対してテスト計測を実行します。

<preImageNo> パラメータとして、0 を指定した場合は、最新の本体ロギング画像に対してテスト計測を実行します。

引数を省略した場合は、カメラから画像を入力したテスト計測となります。

パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。

パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定しても、Illegal function call のエラーは発生しません。

整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。

文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。

マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、BUSY 信号などの計測中信号が ON した計測禁止状態で実行してください。（詳細は、「5-1-4 状態遷移と実行タイミング (P.5-12)」を参照してください。）
- *MCRINIT に記述しないでください。

・使用例

指定した登録画像を計測画像として読み込みます。

```
Rem 登録画像の保存先パス取得
USBPATH$ = ApplicationPath$(2)
filename$ = argumentstring$(0)
FilePATH$ = USBPATH$ + "RegisteredImage¥" + filename$ + ".ifz"

Rem 計測画像として読み込み
Testmeasure FilePATH$
```

・使用可能なモジュール
シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン
バージョン 5.40 以降

・関連項目

「ApplicationPath$ (P.8-26)」
「MeasureStart (P.8-182)」
「Remeasure (P.8-220)」
「Measure (P.8-176)」
「MeasureStop (P.8-183)」

Time$

内蔵クロックの時刻を読み出します。

・書式
Time$

・パラメータ
なし

・戻り値
文字列型の時刻の値を返します。
時刻の値は、内蔵クロックの時刻を時（HH）、分（MM）、秒（SS）それぞれコロン（:）で区切った文字列です。時分秒それぞれの範囲は以下のとおりです。
- 時（HH）：00～23
- 分（MM）：00～59
- 秒（SS）：00～59

・解説
内蔵クロックの時刻を時分秒の文字列として読み出します。
内蔵クロックの設定はシステム設定の日時設定で設定します。（詳細は、『画像処理システム FH/FHV シリーズ ユーザーズマニュアル(SDNB-719)』の「日付と時刻を合わせる[その他]」を参照してください。）
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
内蔵クロックの時刻を読み出して、システム状態監視コンソールウィンドウに出力します。

```
Dim NOW$(2)

Rem 内蔵クロックの時刻を読み出します
NOW$ = Time$

Rem 読み出した時刻の書式を変更します
For I&=0 To 2
   NOW$(I&) = Piece$(NOW$, ":", I&+1, I&+1)
Next

Rem 書式を変更した時刻をシステム状態監視コンソールウィンドウに出力します
Print NOW$(0) + "時" + NOW$(1) + "分" + NOW$(2) + "秒"
```

結果は以下のとおりになります。

```
01 時 23 分 45 秒
```

・使用可能なモジュール
ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「Date$ (P.8-62)」
「Mid$ (P.8-186)」
「Print (P.8-202)」
「GetSystemData (P.8-133)」
「Piece$ (P.8-200)」
「SetSystemData (P.8-271)」

Timer

経過時間を取得します。

・書式
Timer(<start>, <mode>)

・パラメータ

パラメータ名	データ型	説明
<start>	実数型	経過時間の測定を開始した StartTimer 関数の戻り値
<mode>	整数型	取得する経過時間の単位 0：ms 単位 1：μs 単位

・戻り値
整数型の StartTimer 関数を実行してからの経過時間の、小数点以下を切り捨てた値を返します。

・解説
<mode> パラメータで指定した単位で、<start> パラメータで指定した開始時間からの経過時間を取得します。
<start> パラメータには、このマクロ関数よりも前に実行する StartTimer 関数の戻り値を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。

マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数や StartTimer 関数自身の処理時間により、取得できる経過時間に誤差が生じることがあります。

・使用例

測定する対象となる処理の処理時間を 4 回測定し、表示します。

```
Dim TIME&(3)


For NUM& = 0 To 3
T# = StartTimer
Rem 測定する対象となる処理を実行します

Rem StartTimer 関数の戻り値を使って経過時間を取得します
TIME&(NUM&) = Timer(T#, 0)
Next

For NUM& = 0 To 3
Print(TIME&(NUM&))
Next
```

StartTimer 関数と Timer 関数の間に、Print 関数などの測定する対象以外の処理を入れないでください。

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「ElapsedTime (P.8-103)」
「StartTimer (P.8-285)」
「Wait (P.8-328)」

TotalJudge

総合判定結果を取得します。

・書式

TotalJudge

・パラメータ

なし

・戻り値

整数型の総合判定結果の値を返します。

- 0：判定なし（未計測）
- 1：判定結果 OK
- -1：判定結果 NG

・解説

計測フローの実行結果である総合判定結果の値を取得します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

ユニットマクロ処理ユニットの*MEASUREDISPG サブルーチンの中で、総合判定結果を取得して画像表示ウィンドウに文字列を表示します。

```
CHARSTRING$ = ""

Rem 総合判定結果を取得します
JG& = TotalJudge

Rem 総合判定結果によって表示する文字列を変更します
If JG& = 1 Then
   CHARSTRING$ = "OK"
Elself JG& = -1 Then
   CHARSTRING$ = "NG"
Elself JG& = 0 Then
   CHARSTRING$ = "NC"
Endif

Rem 文字列を表示します
DrawTextG CHARSTRING$, 100, 100, 0
```

・使用可能なモジュール

ユニット演算マクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「DrawTextG (P.8-99)」　「SetUnitJudge (P.8-277)」

TransformAngle

角度の値にキャリブレーション結果や位置ずれ修正量を反映します。

・書式

TransformAngle <unitNo>, <imageNo>, <mode>, <srcAngle>, <destAngle>

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	変換するデータを持つ処理ユニットの処理ユニット番号（0～（現在シーンの処理ユニット数 -1)）
<imageNo>	整数型	変換するデータを持つ処理ユニットの計測画像番号（常に 0）
<mode>	整数型	変換種別 0：画像変換後 → 画像変換前 1：画像変換前 → 画像変換後 10：画像座標系 → キャリブレーション後
<srcAngle>	実数型	変換するデータを持つ処理ユニットから取得した変換前の角度
<destAngle>	実数型	変換後の角度

・戻り値

なし

・解説

<unitNo> パラメータで指定した処理ユニットの、<imageNo> パラメータで指定した画像上で、<srcAngle> パラメータで指定した角度の値に、<mode> パラメータで指定した変換内容を反映して値を変換します。
<mode> パラメータには、0、1、10 のいずれかの値を指定してください。0、1、10 以外の値を指定した場合は、このマクロ関数を実行したときの動作が不定になります。
<destAngle> パラメータには、変換後の角度の値を格納する変数を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

処理ユニット番号 2 のサーチ処理ユニットで計測したキャリブレーション適用前の計測角度の値を、キャリブレーション適用後の角度の値に変換します。「計測角度」は外部参照データ識別名"angle" です。

```
Rem 計測結果を取得します
GetUnitData 2, "angle", BEFOREANGLE#

Rem キャリブレーション適用後の値に変換します
TransformAngle 2, 0, 10, BEFOREANGLE#, AFTERANGLE#
```

・使用可能なモジュール

ユニット演算マクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「GetUnitData (P.8-138)」
「TransformDist (P.8-303)」
「TransformXY (P.8-305)」
「Ut (P.8-320)」
「TransformArea (P.8-302)」
「TransformLine (P.8-304)」
「UnitNo (P.8-318)」

TransformArea

面積の値にキャリブレーション結果や位置ずれ修正量を反映します。

・書式

TransformArea <unitNo>, <imageNo>, <mode>, <srcArea>, <destArea>

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	変換するデータを持つ処理ユニットの処理ユニット番号（0～（現在シーンの処理ユニット数 -1)）
<imageNo>	整数型	変換するデータを持つ処理ユニットの計測画像番号（常に 0）
<mode>	整数型	変換種別 0：画像変換後 → 画像変換前 1：画像変換前 → 画像変換後 10：画像座標系 → キャリブレーション後
<srcArea>	実数型	変換するデータを持つ処理ユニットから取得した変換前の面積
<destArea>	実数型	変換後の面積

・戻り値

なし

・解説

<unitNo> パラメータで指定した処理ユニットの、<imageNo> パラメータで指定した画像上で、<srcArea> パラメータで指定した面積の値に、<mode> パラメータで指定した変換内容を反映して値を変換します。
<mode> パラメータには、0、1、10 のいずれかの値を指定してください。0、1、10 以外の値を指定した場合は、このマクロ関数を実行したときの動作が不定になります。
<destArea> パラメータには、変換後の面積の値を格納する変数を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

処理ユニット番号 2 のラベリング処理ユニットで計測したキャリブレーション適用前の面積の値を、キャリブレーション適用後の面積の値に変換します。「面積」は外部参照データ識別名"area" です。

```
Rem 計測結果を取得します
GetUnitData 2, "area", BEFOREAREA#

Rem キャリブレーション適用後の値に変換します
TransformArea 2, 0, 10, BEFOREAREA#, AFTERAREA#
```

・使用可能なモジュール

ユニット演算マクロ ／ ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「GetUnitData (P.8-138)」
「TransformAngle (P.8-301)」
「TransformDist (P.8-303)」
「TransformLine (P.8-304)」
「TransformXY (P.8-305)」
「UnitNo (P.8-318)」
「Ut (P.8-320)」

TransformDist

距離の値にキャリブレーション結果や位置ずれ修正量を反映します。

・書式
TransformDist <unitNo>, <imageNo>, <mode>, <srcDist>, <destDist>

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	変換するデータを持つ処理ユニットの処理ユニット番号（0～（現在シーンの処理ユニット数 -1)）
<imageNo>	整数型	変換するデータを持つ処理ユニットの計測画像番号（常に 0）
<mode>	整数型	変換種別 0：画像変換後 → 画像変換前 1：画像変換前 → 画像変換後 10：画像座標系 → キャリブレーション後
<srcDist>	実数型	変換するデータを持つ処理ユニットから取得した変換前の距離
<destDist>	実数型	変換後の距離

・戻り値
なし

・解説
<unitNo> パラメータで指定した処理ユニットの、<imageNo> パラメータで指定した画像上で、<srcDist> パラメータで指定した距離の値に、<mode> パラメータで指定した変換内容を反映して値を変換します。
<mode> パラメータには、0、1、10 のいずれかの値を指定してください。0、1、10 以外の値を指定した場合は、このマクロ関数を実行したときの動作が不定になります。
<destDist> パラメータには、変換後の距離の値を格納する変数を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
処理ユニット番号 2 のスキャンエッジ幅処理ユニットで計測したキャリブレーション適用前の平均幅の距離の値を、キャリブレーション適用後の距離の値に変換します。「平均幅」は外部参照データ識別名"width_ave" です。

```
Rem 計測結果を取得します
GetUnitData 2, "width_ave", BEFOREDIST#

Rem キャリブレーション適用後の値に変換します
TransformDist 2, 0, 10, BEFOREDIST#, AFTERDIST#
```

・使用可能なモジュール
ユニット演算マクロ ／ ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「GetUnitData (P.8-138)」
「TransformArea (P.8-302)」
「TransformXY (P.8-305)」
「Ut (P.8-320)」
「TransformAngle (P.8-301)」
「TransformLine (P.8-304)」
「UnitNo (P.8-318)」

TransformLine

直線成分の値にキャリブレーション結果や位置ずれ修正量を反映します。

・書式
TransformLine <unitNo>, <imageNo>, <mode>, <srcA>, <srcB>, <srcC>, <destA>, <destB>, <destC>

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	変換するデータを持つ処理ユニットの処理ユニット番号（0～（現在シーンの処理ユニット数 -1））
<imageNo>	整数型	変換するデータを持つ処理ユニットの計測画像番号（常に 0）
<mode>	整数型	変換種別 0：画像変換後 → 画像変換前 1：画像変換前 → 画像変換後 10：画像座標系 → キャリブレーション後
<srcA>	実数型	変換するデータを持つ処理ユニットから取得した変換前の直線成分 A
<srcB>	実数型	変換するデータを持つ処理ユニットから取得した変換前の直線成分 B
<srcC>	実数型	変換するデータを持つ処理ユニットから取得した変換前の直線成分 C
<destA>	実数型	変換後の直線成分 A
<destB>	実数型	変換後の直線成分 B
<destC>	実数型	変換後の直線成分 C

・戻り値
なし

・解説
<unitNo> パラメータで指定した処理ユニットの、<imageNo> パラメータで指定した画像上で、<srcA> パラメータ、<srcB> パラメータ、<srcC> パラメータで指定した直線の直線成分に、<mode> パラメータで指定した変換内容を反映して値を変換します。
<mode> パラメータには、0、1、10 のいずれかの値を指定してください。0、1、10 以外の値を指定した場合は、このマクロ関数を実行したときの動作が不定になります。

<destA> パラメータ、<destB> パラメータ、<destC> パラメータには、変換後の直線成分の値を格納する変数をそれぞれ指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
処理ユニット番号 2 のスキャンエッジ位置処理ユニットで計測したキャリブレーション適用前の直線の直線成分の値を、キャリブレーション適用後の直線成分の値に変換します。「直線成分 A」は外部参照データ識別名"coefficientA"、「直線成分 B」は外部参照データ識別名"coefficientB"、「直線成分 C」は外部参照データ識別名"coefficientC" です。

```
Rem 計測結果を取得します
GetUnitData 2, "coefficientA", BEFOREA#
GetUnitData 2, "coefficientB", BEFOREB#
GetUnitData 2, "coefficientC", BEFOREC#

Rem キャリブレーション適用後の値に変換します
TransformLine 2, 0, 10, BEFOREA#, BEFOREB#, BEFOREC#, AFTERA#, AFTERB#, AFTERC#
```

・使用可能なモジュール
ユニット演算マクロ ／ ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「GetUnitData (P.8-138)」
「TransformArea (P.8-302)」
「TransformXY (P.8-305)」
「Ut (P.8-320)」
「TransformAngle (P.8-301)」
「TransformDist (P.8-303)」
「UnitNo (P.8-318)」

TransformXY

座標の値にキャリブレーション結果や位置ずれ修正量を反映します。

・書式
TransformXY <unitNo>, <imageNo>, <mode>, <srcX>, <srcY>, <destX>, <destY>

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	変換するデータを持つ処理ユニットの処理ユニット番号（0～（現在シーンの処理ユニット数 -1)）
<imageNo>	整数型	変換するデータを持つ処理ユニットの計測画像番号（常に 0）
<mode>	整数型	変換種別 0：画像変換後 → 画像変換前 1：画像変換前 → 画像変換後 10：画像座標系 → キャリブレーション後
<srcX>	実数型	変換するデータを持つ処理ユニットから取得した変換前の X 座標

パラメータ名	データ型	説明
<srcY>	実数型	変換するデータを持つ処理ユニットから取得した変換前の Y 座標
<destX>	実数型	変換後の X 座標
<destY>	実数型	変換後の Y 座標

・戻り値
なし

・解説
<unitNo> パラメータで指定した処理ユニットの、<imageNo> パラメータで指定した画像上で、<srcX> パラメータと<srcY> パラメータで指定した座標の値に、<mode> パラメータで指定した変換内容を反映して値を変換します。
<mode> パラメータには、0、1、10 のいずれかの値を指定してください。0、1、10 以外の値を指定した場合は、このマクロ関数を実行したときの動作が不定になります。
<destX> パラメータと<destY> パラメータには、変換後の座標の値を格納する変数を指定してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
処理ユニット番号 2 のサーチ処理ユニットで計測したキャリブレーション適用前の計測座標の値を、キャリブレーション適用後の計測座標の値に変換します。「計測座標 X」は外部参照データ識別名 "positionX"、「計測座標 Y」は外部参照データ識別名"positionY" です。

```
Rem 計測結果を取得します
GetUnitData 2, "positionX", BEFOREX#
GetUnitData 2, "positionY", BEFOREY#

Rem キャリブレーション適用後の値に変換します
TransformXY 2, 0, 10, BEFOREX#, BEFOREY#, AFTERX#, AFTERY#
```

・使用可能なモジュール
ユニット演算マクロ ／ ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「GetUnitData (P.8-138)」
「TransformArea (P.8-302)」
「TransformLine (P.8-304)」
「Ut (P.8-320)」
「TransformAngle (P.8-301)」
「TransformDist (P.8-303)」
「UnitNo (P.8-318)」

Try Catch End Try

エラーの発生を検知して例外処理します。

・書式

Try
<statement>
Catch
<exceptionStatement>]
End Try

・パラメータ

パラメータ名	データ型	説明
<statement>	-	エラーを発生させる可能性があるステートメント
<exceptionstatement>	-	エラーが発生したときに実行するステートメント

・戻り値

なし

・解説

<statement> パラメータで指定した Try ブロックのステートメントを実行して、エラーが発生した場合に<exceptionStatement> パラメータで指定した Catch ブロックのステートメントを実行します。
Try ブロックの中のすべてのステートメントを実行してエラーが発生しなかった場合は、Catch ブロックのステートメントを実行せず処理を終了します。
Try ブロックの中で発生したエラーのエラー番号やエラーが発生したマクロ関数名を取得する場合は、Catch ブロックの中で Errno 関数または Errcmnd$ 関数を使用してください。（参照：「Errno (P.8-107)」、「Errcmnd$ (P.8-106)」）
Catch ブロックの中でエラーが発生した場合は、エラーを検知できません。Catch ブロックの中でエラーを発生させる可能性がある場合は、Try Catch End Try 関数をネストしてエラーの発生を検知してください。
ステートメントの中で Goto 関数を使用して Try ブロックまたは Catch ブロックの中から外に制御を移したり、外から中に制御を移したりした場合は、予期せぬ動作になる可能性があります。
Try 文、End Try 文のいずれかを記述しなかった場合は、記述の内容によって CATCH without TRY、END TRY without TRY のいずれかのエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

ユニットマクロの*MEASUREPROC サブルーチンの中で、Try Catch End Try 関数でエラーの発生を検知してエラー番号を取得します。

```
*MEASUREPROC

   Try
      WORK& = 0
      SUMM& = 100 + 200 + 300
      ANS& = SUMM& / WORK&
   Catch
      If Errno = 11 Then
         Rem エラー番号とエラーの内容をシステム状態監視コンソールウィンドウに出力します
         Print "Error Number = " + Str$(Errno) + ", Division by Zero"
      Endif
   End Try

Return
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「Errcmnd$ (P.8-106)」
「Errno (P.8-107)」
「If Then Elseif Else Endif (P.8-144)」
「Print (P.8-202)」
「Str$ (P.8-287)」

UCase$

小文字を大文字に変換します。

・書式

UCase$(<string>)

・パラメータ

パラメータ名	データ型	説明
<string>	文字列型	大文字に変換するアルファベットを含む文字列

・戻り値

文字列型のアルファベット大文字に変換した値を返します。

・解説

<string> パラメータで指定した文字列に含まれるアルファベットの小文字を大文字に変換します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、256 文字め以降を切り捨てた 255 文字の文字列として扱われます。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

小文字を大文字に変換します。

```
CHARA1$ = "Measurement Result = 100.0(OK)"

Rem 文字列に含まれるアルファベットの小文字を大文字に変換します
CHARA2$ = UCase$(CHARA1$)
```

結果は以下のとおりになります。

```
CHARA2$ = "MEASUREMENT RESULT = 100.0(OK)"
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 4.20 以降

・関連項目

「Asc (P.8-31)」
「Chr$ (P.8-41)」
「Hex$ (P.8-142)」
「LCase$ (P.8-162)」
「Left$ (P.8-163)」
「Len (P.8-164)」
「Mid$ (P.8-186)」
「Piece$ (P.8-200)」
「Right$ (P.8-223)」
「Str$ (P.8-287)」
「Str2$ (P.8-288)」
「Val (P.8-321)」

UnitCount

処理ユニットの登録数を取得します。

・書式
UnitCount

・パラメータ
なし

・戻り値
整数型の処理ユニット登録数の値を返します。

・解説
現在のシーンに登録されている処理ユニットの数を取得します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
計測フローの末尾にサーチの処理ユニットを追加します。

```
Rem 現在のフローに登録されている処理ユニット数を取得します
UNUM& = UnitCount

Rem 処理項目の識別子を設定します
IDENT$ = "Search"

Rem 処理項目サーチをフローの最後に追加します
AssignUnit UNUM& , IDENT$
```

・使用可能なモジュール
シーン制御マクロ ／ 通信コマンドマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「AssignUnit (P.8-32)」
「CheckUnit (P.8-40)」
「CopyUnit (P.8-54)」
「DeleteUnit (P.8-66)」
「InsertUnit (P.8-152)」
「MeasureStart (P.8-182)」
「MeasureStop (P.8-183)」
「MoveUnit (P.8-190)」

UnitData

処理ユニットの数値データを取得します。

・書式

UnitData(<unitNo>, <dataNo>)
UnitData(<unitNo>, <dataIdent>)

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	処理ユニット番号（0～（現在シーンの処理ユニット数 -1)）
<dataNo>	整数型	取得する処理ユニットデータの外部参照データ番号（参照：『画像処理システム FH/FHV シリーズ 処理項目機能リファレンスマニュアル（SDNB-713)』）
<dataIdent>	文字列型	取得する処理ユニットデータのデータ識別名

・戻り値

整数型または実数型の処理ユニットデータの値を返します。
数値ではないデータを取得する場合は、取得対象のデータを数値に変換した値を返します。

・解説

<unitNo> パラメータで指定した処理ユニットの持つ、<dataNo> パラメータで指定した外部参照データ番号のデータを取得します。 <dataNo> パラメータの代わりに<dataIdent> パラメータを指定してデータを取得することもできます。
数値データ以外のデータを取得する場合は、UnitData$ 関数または GetUnitData 関数をご利用ください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

処理ユニット番号 5 のサーチ処理ユニットの計測 X 座標の値を取得します。計測 X 座標は外部参照データ番号 6、外部参照データ識別名"X" です。

```
SEARCH# = UnitData(5, 6)

Rem 6 の代わりに"X" を指定しても同じ結果が得られます
SEARCH# = UnitData(5, "X")
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「GetUnitData (P.8-138)」
「UnitData$ (P.8-312)」
「UnitNo (P.8-318)」
「SetUnitData (P.8-275)」
「UnitData2 (P.8-313)」
「Ut (P.8-320)」

UnitData$

処理ユニットの文字列データを取得します。

・書式

UnitData$(<unitNo>, <dataNo>)
UnitData$(<unitNo>, <dataIdent>)

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	処理ユニット番号（0～（現在シーンの処理ユニット数 -1)）
<dataNo>	整数型	取得する処理ユニットデータの外部参照データ番号（参照：『画像処理システム FH/FHV シリーズ 処理項目機能リファレンスマニュアル（SDNB-713)』）
<dataIdent>	文字列型	取得する処理ユニットデータのデータ識別名

・戻り値

文字列型の処理ユニットデータの値を返します。
文字列ではないデータを取得する場合は、取得対象のデータを文字列に変換した値を返します。

・解説

<unitNo> パラメータで指定した処理ユニットの持つ、<dataNo> パラメータで指定した外部参照データ番号のデータを取得します。 <dataNo> パラメータの代わりに<dataIdent> パラメータを指定してデータを取得することもできます。
文字列データ以外のデータを取得する場合は、UnitData 関数または GetUnitData 関数をご利用ください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

処理ユニット番号 5 の 2 次元コード処理ユニットのデコード文字列を取得します。デコード文字列は外部参照データ番号 7、外部参照データ識別名"decodeCharStr" です。

```
DECODECHAR$ = UnitData$(5, 7)

Rem 7 の代わりに"decodeCharStr" を指定しても同じ結果が得られます
DECODECHAR$ = UnitData$(5, "decodeCharStr")
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「GetUnitData (P.8-138)」
「UnitData (P.8-311)」
「UnitNo (P.8-318)」
「SetUnitData (P.8-275)」
「UnitData2 (P.8-313)」
「Ut (P.8-320)」

UnitData2

処理ユニットの描画座標データを取得します。

・書式

UnitData2(<unitNo>, <dataNo>)
UnitData2(<unitNo>, <dataIdent>)

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	処理ユニット番号（0～（現在シーンの処理ユニット数 -1)）
<dataNo>	整数型	取得する処理ユニットデータの外部参照データ番号（参照：『画像処理システム FH/FHV シリーズ 処理項目機能リファレンスマニュアル（SDNB-713)』）
<dataIdent>	文字列型	取得する処理ユニットデータのデータ識別名

・戻り値

整数型または実数型の処理ユニットデータの値を返します。

・解説

<unitNo> パラメータで指定した処理ユニットの持つ、<dataNo> パラメータで指定した外部参照データ番号のデータを取得します。 <dataNo> パラメータの代わりに<dataIdent> パラメータを指定してデータを取得することもできます。
このマクロ関数では、キャリブレーションなどで変換される前の計測座標値を取得できます。画像表示ウィンドウに計測結果を表示するための描画座標を取得する場合に使用してください。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

ユニットマクロ処理ユニットの*MEASUREDISPG サブルーチンの中で、処理ユニット番号 5 のサーチ処理ユニットの計測 X 座標と Y 座標の値を取得して、画像上の座標にカーソルを表示します。計測 X 座標は外部参照データ番号 6 で外部参照データ識別名"X"、計測 Y 座標は外部参照データ番号 7 で外部参照データ識別名"Y" です。
サーチ処理ユニットの「キャリブレーション」の設定が「ON」でも、このマクロ関数を使用することでキャリブレーションの設定内容を気にすることなく描画座標を取得できます。

```
SEARCHX& = UnitData2(5, 6)
SEARCHY& = UnitData2(5, 7)

Rem 6 の代わりに"X" を指定しても同じ結果が得られます
SEARCHX& = UnitData2(5, "X")
SEARCHY& = UnitData2(5, "Y")

Rem 位置ずれ修正前の座標にカーソルを表示します
DrawCursor SEARCHX&, SEARCHY&, 0, UnitNo
```

・使用可能なモジュール

ユニット演算マクロ ／ ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「GetUnitData (P.8-138)」
「UnitData (P.8-311)」
「UnitNo (P.8-318)」
「SetUnitData (P.8-275)」
「UnitData$ (P.8-312)」
「Ut (P.8-320)」

UnitInfo

処理ユニットの情報を取得します。

・書式
UnitInfo(<unitNo>, <kind>)

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	情報を取得する処理ユニットの処理ユニット番号（0～（現在シーンの処理ユニット数-1））
<kind>	整数型	情報種別 0：処理項目種別 処理項目の種別を表す番号です。以下の値を取得できます。 • 0：検査・計測する（計測） • 1：画像を取り込む（画像入力） • 2：画像を補正する（画像補正） • 3：検査・計測を補助する（計測補助） • 4：処理を分岐する（分岐制御） • 5：結果を外部へ出力する（結果出力） • 6：結果を画面へ表示する（結果表示） 1：設定データ構造体サイズ 設定データ構造体のサイズです。単位はバイトです。 2：計測データ構造体サイズ 計測データ構造体のサイズです。単位はバイトです。 3：制御データ構造体サイズ 制御データ構造体のサイズです。単位はバイトです。 4：図形データ最大個数 処理ユニットが持つ図形データの最大個数です。 5：モデルデータ最大個数 処理ユニットが持つモデルデータの最大個数です。 6：画像データ最大個数 処理ユニットが持つ画像データの最大個数です。 7：内包処理ユニット最大個数 処理ユニットが内部に持つ（内包する）処理ユニットの最大個数です。 8：カメラ設定有効有無 計測初期化などのタイミングで処理ユニットがカメラ設定を更新するかどうかを表します。カメラ画像入力などの画像入力処理ユニットでは、「1」が返ります。 • 0：カメラ設定無効 • 1：カメラ設定有効

パラメータ名	データ型	説明
		9：処理ユニット計測処理並列実行可否 計測実行時に処理ユニットを計測フロー上で並列実行できるかどうかを表します。並列化処理に対応した処理ユニットでは、「1」が返ります。 • 0：並列実行できない • 1：並列実行できる 並列化処理については、『画像処理システム FH/FHV シリーズ ユーザーズマニュアル (SDNB-719)』の「並列処理機能」を参照してください。

・戻り値

整数型の処理ユニットの情報の値を返します。情報が存在しない場合は、-1 を返します。

・解説

<unitNo> パラメータで指定した処理ユニットの、<kind> パラメータで指定した情報を取得します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

処理ユニット番号 2 の処理ユニットにモデル登録が可能かどうかチェックします。

```
If UnitInfo(2, 5) > 0 Then
   Rem モデル登録処理を記述します
Endif
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「UnitNo (P.8-318)」　「Ut (P.8-320)」

UnitItemIdent$

処理ユニットの処理項目識別名を取得します。

・書式

UnitItemIdent$(<unitNo>)

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	処理項目識別名「9-5 処理項目識別名一覧 (P.9-64)」を取得する処理ユニットの処理ユニット番号 (0～(現在シーンの処理ユニット数-1))

・戻り値

文字列型の処理項目識別名の値を返します。

・解説

<unitNo> パラメータで指定した処理ユニットの処理項目識別名を取得します。指定した処理ユニットが計測フローに未登録の場合は、ヌル文字列（""）を返します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定しても、Illegal function call のエラーは発生しません。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

シーン制御マクロで、計測フローに登録されているサーチ処理ユニットを検索して、判定条件である相関値下限値を「70」に変更します。サーチの相関値下限値は、外部参照データ番号 143 です。

```
Rem 処理ユニットの登録数を取得します
COUNT& = UnitCount

Rem サーチ処理ユニットを検索します
For I&=0 To COUNT&-1
   If UnitItemIdent$(I&) = "Search" Then
      Rem サーチ処理ユニットの判定条件である相関値下限値を変更します
      SetUnitData I&, 143, 70
   Endif
Next
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「SetUnitData (P.8-275)」
「UnitCount (P.8-310)」
「UnitNo (P.8-318)」
「Ut (P.8-320)」

UnitJudge

処理ユニットの判定結果を取得します。

・書式

UnitJudge(<unitNo>)

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	判定結果を取得する処理ユニットの処理ユニット番号（0～（現在シーンの処理ユニット数-1））

・戻り値

整数型の判定結果の値を返します。

- 0：判定なし（未計測）
- 1：判定結果 OK
- -1：判定結果 NG
- -10：判定結果エラー（画像不整合）
- -11：判定結果エラー（モデル未登録）
- -12：判定結果エラー（メモリ不足）
- -20：判定結果エラー（その他のエラー）

・解説

<unitNo> パラメータで指定した処理ユニットの判定結果を取得します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

処理ユニット番号 5 の処理ユニットの判定結果を取得します。

```
JUDGE& = UnitJudge(5)
```

・使用可能なモジュール

ユニット演算マクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「GetUnitData (P.8-138)」
「TotalJudge (P.8-300)」
「Ut (P.8-320)」
「SetUnitJudge (P.8-277)」
「UnitNo (P.8-318)」

UnitNo

処理ユニット番号を取得します。

・書式
UnitNo

・パラメータ
なし

・戻り値
整数型の処理ユニット番号の値を返します。

・解説
当該処理ユニットの処理ユニット番号を取得します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。

・使用上の注意
なし

・使用例
当該処理ユニットのひとつ前に登録されている処理ユニットの判定結果を取得します。

```
Rem 当該処理ユニットの処理ユニット番号を取得します
UNO& = UnitNo

Rem ひとつ前に登録されている処理ユニットの判定結果を取得します
JUDGE& = UnitJudge(UNO& - 1)
```

・使用可能なモジュール
ユニット演算マクロ ／ ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「UnitJudge (P.8-317)」　「Ut (P.8-320)」

UnitTitle$

処理ユニットのタイトル名を取得します。

・書式

UnitTitle$(<unitNo>)

・パラメータ

パラメータ名	データ型	説明
<unitNo>	整数型	タイトル名を取得する処理ユニットの処理ユニット番号（0～（現在シーンの処理ユニット数-1））

・戻り値

文字列型のタイトル名の値を返します。

・解説

<unitNo> パラメータで指定した処理ユニットのタイトル名を取得します。
タイトル名は、言語設定にしたがった言語で取得できます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

シーン制御マクロで、計測フローに登録されているタイトル名"Bolt search" のサーチ処理ユニットを検索して、判定条件である相関値下限値を「70」に変更します。サーチの相関値下限値は、外部参照データ番号 143 です。

```
Rem 処理ユニットの登録数を取得します
COUNT& = UnitCount

Rem タイトル名が"Bolt search" の処理ユニットを検索します
For I&=0 To COUNT&-1
   If UnitTitle$(I&) = "Bolt search" Then
      Rem 判定条件である相関値下限値を変更します
      SetUnitData I&, 143, 70
   Endif
Next
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「SetUnitData (P.8-275)」
「SetUnitTitle (P.8-278)」
「UnitCount (P.8-310)」
「UnitNo (P.8-318)」
「Ut (P.8-320)」

Ut

ユニットラベルを指定して処理ユニット番号を取得します。

・書式

Ut(<unitLabel>)

・パラメータ

パラメータ名	データ型	説明
<unitLabel>	文字列型	処理ユニットのユニットラベル

・戻り値

整数型の処理ユニット番号の値を返します。

・解説

<unitLabel> パラメータで指定したユニットラベルが設定された処理ユニットの処理ユニット番号を取得します。
ユニットラベルが分かれば処理ユニット番号を取得できるので、計測フローを変更して処理ユニット番号が変わっても、プログラムを変更する必要がありません。
ユニットラベルは、あらかじめシーン制御マクロツールで設定してください。(参照:「3-1-4 シーン制御マクロツールの設定画面の見かたと設定方法 (P.3-12)」)
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

シーン制御マクロツールで、処理ユニット番号 10 番のサーチ処理ユニットに設定したユニットラベル「PositionSearch」の処理ユニット番号を取得して、判定結果を取得します。

```
Rem あらかじめシーン制御マクロツールで設定しておいたユニットラベルを指定して処理ユニット番号を取得
します
UNITNO& = Ut("PositionSearch")

Rem 取得した処理ユニット番号をもとに処理ユニットの判定結果を取得します
JG& = UnitJudge(UNITNO&)
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 5.20 以降

・関連項目

「UnitJudge (P.8-317)」　「UnitNo (P.8-318)」

Val

数字文字列を数値に変換します。

・書式

Val(<string>)

・パラメータ

パラメータ名	データ型	説明
<string>	文字列型	数値に変換する数字文字列

・戻り値

実数型の数値の値を返します。

・解説

<string> パラメータで指定した数字文字列を数値に変換します。
Val 関数の逆関数には Str$ 関数があります。Str$ 関数は、指定した数値を数字文字列に変換します。
<string> パラメータで指定する数字文字列には、先頭文字として"+"、"-"、"." または"0" から"9" までの半角数字文字を指定してください。先頭文字にこれら以外の文字が指定された場合は 0 を返します。
<string> パラメータで指定した数字文字列に数値に変換できない文字が含まれていた場合は、数値に変換できない文字の直前までの文字を数値に変換します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定しても、エラーは発生しません。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

数字文字列を数値に変換します。

```
VALUE1# = Val(″123.456″)
VALUE2# = Val(″-123.456″)
VALUE3# = Val(″.123″)
VALUE4# = Val(″-.456″)
VALUE5# = Val(″123″+″.″+″456″)
VALUE6# = Val(″123+456″)
```

結果は以下のとおりになります。

```
VALUE1# = 123.456
VALUE2# = -123.456
VALUE3# = 0.123
VALUE4# = -0.456
VALUE5# = 123.456
VALUE6# = 123
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「Asc (P.8-31)」
「Hex$ (P.8-142)」
「Left$ (P.8-163)」
「Mid$ (P.8-186)」
「Right$ (P.8-223)」
「Str2$ (P.8-288)」
「Chr$ (P.8-41)」
「LCase$ (P.8-162)」
「Len (P.8-164)」
「Piece$ (P.8-200)」
「Str$ (P.8-287)」
「UCase$ (P.8-309)」

VarList

システム状態監視コンソールウィンドウに指定した変数の値の一覧を出力します。

・書式

VarList [<variableName>]

・パラメータ

パラメータ名	データ型	説明
<variableName>	文字列型	出力対象の変数の変数名 このパラメータは省略可能です。 パラメータを指定しない場合は、現在のスコープで使用されている変数の値が一覧で出力されます。

・戻り値

なし

・解説

<variableName> パラメータで指定した変数の値を、システム状態監視コンソールウィンドウに出力します。
このマクロ関数では、配列変数の各要素の値は出力できません。配列変数の各要素の値を出力する場合は、「? AA&(5)」のように、「?」と半角スペースに続けて要素番号を指定した配列変数の各要素を指定してください。
<variableName> パラメータには、ワイルドカードとして*（文字列パターン）と?（1 文字）を使用することができます。
ワイルドカードでは次のような指定が可能です。

*	すべての変数を指定します
???	3 文字の変数名の変数を指定します
A*	A で始まる変数名の変数を指定します
*A*A*A*	A を 3 文字以上含む変数名の変数を指定します
????*	4 文字以上の変数名の変数を指定します

パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、Stop 関数でプログラムの実行を中断している状態で使用してください。
- VarList コマンドによる変数の表示は、Option Explicit コマンドの使用によって振る舞いが変わります。
  Option Explicit コマンド使用時、参照変数を設定し、VarList コマンドを実行しても表示されません。Dim 関数で定義後、VarList コマンドを実行すると表示されます。

Option Explicit コマンド不使用時、参照変数を設定し、VarList コマンド実行した時点で表示されます。

・使用例
処理ユニット番号 1 のユニットマクロ処理ユニットの変数として使用している AA&、AB#、BB&、ABC$、DEF# のうち、頭文字が「A」で型識別子含めて 3 文字の変数 AA&、AB$ の値の一覧を出力します。

```
Macro(U1)>VarList "A??"
AA&=123
AB#=123.456
```

・使用可能なモジュール
ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 5.20 以降

・関連項目

「Cls (P.8-48)」
「Cont (P.8-49)」
「Debug (P.8-63)」
「DebugPrint (P.8-64)」
「Dim (P.8-67)」
「List (P.8-167)」
「Option Explicit (P.8-198)」
「Print (P.8-202)」
「SetStop (P.8-270)」
「SetVar (P.8-281)」
「Stop (P.8-286)」

VarPop

一時的に退避した変数の値を復元します。

・書式
VarPop

・パラメータ
なし

・戻り値
なし

・解説
直近の VarPush 関数で退避したすべての変数の値を復元します。
VarPush 関数を複数回数実行している場合は、あとから実行した VarPush 関数で退避した変数の値から順に復元します。
退避した変数の値がない場合は、Internal error のエラーが発生します。

・使用上の注意
なし

・使用例
サブルーチンの中で使用する変数をローカル変数の様に扱います。

```
*EXPA

   Rem 変数の現在値を表示します
   Print A&, B&, C&, D#, E#

   Rem EXPA で内部変数のように使いたい変数の現在値を退避させます
   VarPush A&, B&, C&, D#, E#

   Rem A&,B&,C&,D#,E# を自由に使用します
   GetUnitData 2, "CR", A&
   GetUnitData 3, "CR", B&
   GetUnitData 4, "CR", C&
   GetUnitData 5, "X", D#
   GetUnitData 6, "Y", E#

   Rem サブルーチンを呼ぶ前の変数の現在値を確認します
   Print A&, B&, C&, D#, E#

   Rem 処理の入れ子を行う場合
   Rem サブルーチン*EXPB の中でも同名の A&、B&、C& という変数を
   Rem 使っていますが、*EXPB の中でも Varpush／Varpop で変数内容
   Rem の退避／復帰を行っているので、変数の現在値が勝手に書換わって
   Rem しまうことはありません
   Gosub *EXPB

   Rem サブルーチンを呼んだ後の変数の現在値を確認します
   Print A&, B&, C&, D#, E#

   Rem サブルーチン EXPA の先頭で退避した変数の現在値を復元します
   VarPop

Return

*EXPB

   Rem EXPB で内部変数のように使いたい変数の現在値を退避させます
   Rem 変数 A&、B&、C&、D#、E# の内容は、サブルーチン*EXPA の先頭で
   Rem 実行した VarPush とは別の領域に保存されるので、以前退避した値
   Rem が無くなることはありません
   Rem VarPush は最大 16 回まで実行することが可能です
   VarPush A&, B&, C&, D#, E#

   Rem A&, B&, C&, D#, E# を自由に使用します
   GetUnitData 2, "X", A&
   GetUnitData 3, "X", B&
   GetUnitData 4, "X", C&
   D# = 3
   E# = 100 / 512

   Rem 変更後の変数の現在値を確認します
   Print A&, B&, C&, D#, E#

   Rem サブルーチン EXPB の先頭で退避した変数の現在値を復元します
   VarPop

Return
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「VarPush (P.8-325)」

VarPush

変数の値を一時的に退避します。

・書式

VarPush <variable>[, <variable>[,..., <variable>]]

・パラメータ

パラメータ名	データ型	説明
<variable>	文字列型	値を退避する変数の変数名

・戻り値

なし

・解説

<variable> パラメータで指定した変数の値を一時的に退避します。退避した変数の値は VarPop 関数を実行して復元してください。(参照：「VarPop (P.8-323)」)

VarPop 関数を実行せずにこのマクロ関数を 17 回以上実行した場合は、Internal error のエラーが発生します。この関数の実行回数が VarPop 関数の実行回数+16 回以内になるように、VarPop 関数を実行して変数の値を復元してください。

パラメータとして、文字列型または配列型の変数を指定した場合は、Type mismatch のエラーが発生します

・使用上の注意

なし

・使用例

サブルーチンの中で使用する変数をローカル変数の様に扱います。

```
*EXPA

   Rem 変数の現在値を表示します
   Print A&, B&, C&, D#, E#

   Rem EXPA で内部変数のように使いたい変数の現在値を退避させます
   VarPush A&, B&, C&, D#, E#

   Rem A&,B&,C&,D#,E# を自由に使用します
   GetUnitData 2, "CR", A&
   GetUnitData 3, "CR", B&
   GetUnitData 4, "CR", C&
   GetUnitData 5, "X", D#
   GetUnitData 6, "Y", E#

   Rem サブルーチンを呼ぶ前の変数の現在値を確認します
   Print A&, B&, C&, D#, E#

   Rem 処理の入れ子を行う場合
   Rem サブルーチン*EXPB の中でも同名の A&、B&、C& という変数を
   Rem 使っていますが、*EXPB の中でも Varpush／Varpop で変数内容
   Rem の退避／復帰を行っているので、変数の現在値が勝手に書換わって
   Rem しまうことはありません
   Gosub *EXPB

   Rem サブルーチンを呼んだ後の変数の現在値を確認します
   Print A&, B&, C&, D#, E#

   Rem サブルーチン EXPA の先頭で退避した変数の現在値を復元します
   VarPop

Return

*EXPB

   Rem EXPB で内部変数のように使いたい変数の現在値を退避させます
   Rem 変数 A&、B&、C&、D#、E# の内容は、サブルーチン*EXPA の先頭で
   Rem 実行した VarPush とは別の領域に保存されるので、以前退避した値
   Rem が無くなることはありません
   Rem VarPush は最大 16 回まで実行することが可能です
   VarPush A&, B&, C&, D#, E#

   Rem A&, B&, C&, D#, E# を自由に使用します
   GetUnitData 2, "X", A&
   GetUnitData 3, "X", B&
   GetUnitData 4, "X", C&
   D# = 3
   E# = 100 / 512

   Rem 変更後の変数の現在値を確認します
   Print A&, B&, C&, D#, E#

   Rem サブルーチン EXPB の先頭で退避した変数の現在値を復元します
   VarPop

Return
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「VarPop (P.8-323)」

VarSave

変数の値をシーンデータに保存します。

・書式

VarSave <variableName>

・パラメータ

パラメータ名	データ型	説明
<variableName>	文字列型	保存する変数名

・戻り値

なし

・解説

<variableName> パラメータで指定した変数の値をシーンデータに保存します。
<variableName> パラメータには、ワイルドカードとして*（文字列パターン）と?（1 文字）を使用することができます。
ワイルドカードでは次のような指定が可能です。

*	すべての変数を指定します
???	3 文字の変数名の変数を指定します
A*	A で始まる変数名の変数を指定します
*A*A*A*	A を 3 文字以上含む変数名の変数を指定します
????*	4 文字以上の変数名の変数を指定します

このマクロ関数で保存した変数の値はシーンデータをロードしたときに読み込まれます。
このマクロ関数を複数回実行した場合は、シーンデータをロードしたときに実行した順に読み込まれます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定しても、Illegal function call のエラーは発生しません。
文字列型のパラメータとして、255 文字を超える文字列を指定した場合は、String too long のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

- このマクロ関数は、*SAVEPROC サブルーチンの中でしか使用できません。それ以外のサブルーチンで使用しようとした場合は、Illegal function call のエラーが発生します。

・使用例

シーン制御マクロの SAVEPROC サブルーチンの中で、独自に管理するバージョン情報をシーンデータに保存します。

```
*SAVEPROC

    Rem バージョン情報を格納する変数を作成し、100 を設定します
    Version& = 100

    Rem バージョン情報を保存します
    VarSave "Version*"

Return
```

・使用可能なモジュール
シーン制御マクロ

・対応バージョン
バージョン 5.20 以降

・関連項目
なし

Wait

指定した時間だけ処理を待機します。

・書式
Wait <time>

・パラメータ

パラメータ名	データ型	説明
<time>	整数型	待機時間（ms）

・戻り値
なし

・解説
<time> パラメータで指定した時間だけ、処理を待機します。
待機中にバックグラウンドで処理が行われる場合は、バックグラウンドでの処理は待機せず実行されます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意
- パラメータとして指定した待機時間と実際の待機時間との間には誤差が生じることがあります。

・使用例
通信コマンドマクロでシーン 2 に切替後、10ms 待機します。

```
Rem シーンを切り替えます
ChangeScene 2

Rem 10ms 待機します
Wait 10
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン
バージョン 3.50 以降

・関連項目

「ChangeScene (P.8-38)」
「StartTimer (P.8-285)」
「ElapsedTime (P.8-103)」
「Timer (P.8-298)」

WritePlcMemory

PLC のメモリ領域に値を書き込みます。

・書式
WritePlcMemory <ioIdent>, <area>, <channelOffset>, <channelCount>, <writeData()>

・パラメータ

パラメータ名	データ型	説明
<ioIdent>	文字列型	使用する通信モジュールの識別名 （参照：「9-4 IO モジュール一覧 (P.9-25)」）
<area>	整数型	書き込み対象のデータ出力エリアのエリア種別番号
<channelOffset>	整数型	命令エリア先頭から書き込み開始するアドレスまでのオフセット
<channelCount>	整数型	書き込むデータのサイズ（チャネル単位）
<writeData()>	整数型配列	書き込むデータ

・戻り値
なし

・解説
<ioIdent> パラメータで指定した通信モジュールを使用して、<area> パラメータで指定した PLC のエリア種別の、<channelOffset> パラメータで指定した値だけオフセットしたアドレスから、<channelCount> パラメータで指定したサイズ分のデータを書き込みます。
このマクロ関数で PLC のメモリ領域に値を書き込む前に、SetPlcData 関数を実行して書き込む値を設定してください。
<writeData()>パラメータには、書き込むデータを格納した 1 次元の整数型配列変数を要素番号を指定せず( )だけ付加して指定してください。
<area>パラメータには、PLC リンクの場合は、システム設定の PLC リンク設定で設定したレジスタの ID を指定してください。Fieldbus の場合は、エリア種別を指定できないため 0 を指定してください。
<channelCount> パラメータには、チャネル単位のサイズを指定してください。整数型は 1 データが 2 チャネル分（4 バイト）のサイズのため、整数の値を 1 個書き込む場合は、<writeData()>パラメータに要素数 1 の配列を用意して、<channelCount>パラメータには 2 を指定してください。
<channelCount>パラメータに、<writeData()>パラメータで指定した配列のサイズより大きなサイズを指定した場合は、Subscript out of range のエラーが発生します。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
パラメータとして存在しない番号、数値、データ型の組み合わせ、値の組み合わせを指定した場合は、Illegal function call のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。
データを書き込む際には、データ出力エリアを指定してください。応答エリアに書き込んだ場合、コマンド処理後に、レスポンスデータが書き込まれるため、上書きされます。

・使用上の注意

- このマクロ関数を使用してデータを書き込む場合は、必ず SetPlcData 関数を使用して書き込むデータの値を設定してください。SetPlcData 関数を使用せずに<writeData()> パラメータに直接値を設定する場合は、正しい値を設定できない可能性があります。

・使用例

通信コマンドマクロで、PLC リンクで接続された PLC に処理ユニット番号 2 のサーチ処理ユニットの計測座標 X と計測座標 Y を書き込みます。計測座標 X は外部参照データ番号 6、計測座標 Y は外部参照データ番号 7 です。

```
IOMODULE$ = "UdpPlcLink"

Rem 計測結果を取得します
GetUnitData 2, 6, X#
GetUnitData 2, 7, Y#

Rem 実数の計測値を 1000 して整数値に変換します
VALUE0& = Int(X# * 1000)
VALUE1& = Int(Y# * 1000)

Rem データ出力エリアの設定を取得します
GetSystemData IOMODULE$, "outputArea", AREA&
GetSystemData IOMODULE$, "outputMemoryAddress", ADDRESS&

Rem 書き込むデータを整数型配列変数に格納します
Dim DATA&(1)
SetPlcData IOMODULE$, DATA&(), 0, 4, VALUE0&
SetPlcData IOMODULE$, DATA&(), 4, 4, VALUE1&

Rem データ出力エリアにデータ（4CH 分）を書き込みます
WritePlcMemory IOMODULE$, AREA&, ADDRESS&, 4, DATA&()
```

・使用可能なモジュール

シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 4.20 以降

・関連項目

「GetPlcData (P.8-126)」
「GetUnitData (P.8-138)」
「ReadPlcMemory (P.8-210)」
「GetSystemData (P.8-133)」
「Int (P.8-153)」
「SetPlcData (P.8-258)」

XOR

2 式の排他的論理和を求めます。

・書式

<expression1> XOR <expression2>

・パラメータ

パラメータ名	データ型	説明
<expression1>	整数型	排他的論理和を求める数式
<expression2>	整数型	

・戻り値

整数型の排他的論理和の値を返します。

・解説

<expression1> パラメータで指定した数式と、<expression2> パラメータで指定した数式のビットごとの排他的論理和を求めます。
<expression1> パラメータと<expression2> パラメータの値が実数型の場合、小数部を四捨五入した値として扱われます。
パラメータとして間違ったデータ型を指定した場合は、Type mismatch のエラーが発生します。
整数型のパラメータとして、-2147483648～2147483647 の範囲を超える値を指定した場合は、Overflow のエラーが発生します。
戻り値の変数への代入や式の中での使用をしない場合は、Syntax error のエラーが発生します。
マクロ関数名の記述ミス、カンマ忘れや各キーワード間の半角スペース忘れなど、間違った書式を指定した場合は、Syntax error のエラーが発生します。

・使用上の注意

なし

・使用例

12 と 31 の排他的論理和を求めます。

```
DATA1& = 12
DATA2& = 31

DATA3& = DATA1& XOR DATA2&
```

結果は以下のとおりになります。

```
DATA3& = 19
```

・使用可能なモジュール

ユニット演算マクロ ／ シーン制御マクロ ／ 通信コマンドマクロ ／ ユニットマクロ

・対応バージョン

バージョン 3.50 以降

・関連項目

「AND (P.8-25)」
「NOT (P.8-191)」
「UnitData (P.8-311)」
「GetUnitData (P.8-138)」
「OR (P.8-199)」

9

マクロリファレンス一覧

9-1 エラー一覧 9-2

9-2 予約一覧 9-5
9-2-1 予約語一覧 9-5
9-2-2 グローバルデータの予約語一覧 9-9

9-3 システムデーター覧 9-10

9-4 IO モジュール一覧 9-25
9-4-1 EtherCAT 9-26
9-4-2 EtherNetIP 9-28
9-4-3 PROFINET 9-31
9-4-4 ParallelIo 9-34
9-4-5 SerialNormal 9-36
9-4-6 SerialPlcLinkM 9-38
9-4-7 SerialPlcLink 9-40
9-4-8 TcpClient 9-43
9-4-9 TcpNormal 9-45
9-4-10 UdpNormal 9-47
9-4-11 UdpPlcLinkM 9-49
9-4-12 UdpPlcLinkY 9-52
9-4-13 UdpPlcLink 9-55
9-4-14 TcpPlcLinkM 9-58
9-4-15 TcpPlcLink 9-61

9-5 処理項目識別名一覧 9-64
9-5-1 検査・計測をする 9-64
9-5-2 画像を取り込む 9-64
9-5-3 画像を補正する 9-65
9-5-4 検査・計測を補助する 9-65
9-5-5 処理を分岐する 9-66
9-5-6 結果を外部へ出力する 9-66
9-5-7 結果を画面へ表示する 9-66

9-6 図形データー覧 9-67
9-6-1 図形データの構造一覧 9-67
9-6-2 図形データの配列要素一覧 9-68

9-7 図形番号一覧 9-70

9-8 モデル番号一覧 9-72

9-9 画像番号一覧 9-74

9-10 サブ画像番号一覧 9-77

9-1 エラー一覧

マクロカスタマイズ機能のプログラムを実行中にエラーが発生した場合、表示されるエラー番号 (ErrNo) やエラーメッセージから原因を絞り込むことができます。エラーメッセージの種類と内容は以下のとおりです。

使用上の注意

エラー番号は、通信コマンドマクロまたは Try～Catch～End Try の中以外でエラーが発生した場合には参照できません。
システム状態監視コンソールウィンドウに表示されたエラーメッセージでエラーの内容を判断してください。

参考

エラーの情報はシステム状態監視コンソールウィンドウに表示されます。

番号	エラーメッセージ	エラー内容	エラー内容対策
1	NEXT without FOR	For がないのに next が使用されています。	For Next 文で For を正しく記述してください。
2	Syntax error	関数の書式や綴りが間違っています。もしくは関数をルールに則らずに使用しています。	マクロ関数リファレンスを確認して関数の書式や綴りを修正してください。
3	RETURN without GOSUB	Gosub がないのに Return が使用されています。	Gosub 関数が存在しないか正しく記述されていないときに発生します。Gosub 関数を正しく記述してください
5	Illegal function call	関数の引数が許容範囲を超えて指定されています。もしくは関数実行時にエラーが発生しています。	マクロ関数の引数が指定可能な範囲を越えているときに発生します。マクロ関数リファレンスを確認して、マクロ関数の引数を適切に設定してください。 マクロ関数を実行できないサブルーチンの実行場所が不適切な場合は、適切なサブルーチンで実行してください。
6	Overflow	演算結果や入力された数値が倍数度実数型の許容範囲を超えています。もしくは文字列の長さが許容範囲を超えています。	倍数度実数型の数値が-1.0E30～1.0E30 の範囲を超えているときに発生します。倍数度実数型の範囲内におさまるよう処理を修正してください。 文字列型の変数や定数の長さが 255 文字を超えているときに発生します。文字列の長さを短くなるよう修正してください。
7	Out of memory	センサコントローラのワークメモリが不足しています。もしくはループ処理のネストが深くなりすぎています。	プログラムで文字列や配列変数を使いすぎているときに発生します。エラーが発生しなくなるまで使用数を減らしてください。 ループ処理などのネスト数が多いときに発生します。エラーが発生しなくなるまでネスト数を減らしてください。
8	Undefined line number	存在しない行番号に遷移しようとしています。	Goto 関数などで、遷移先の行番号が存在しないときに発生します。存在する行番号を指定するか、行番号ではなく<ラベル>を使用して遷移先を指定してください。
9	Subscript out of range	配列で定義した数以上の要素を使用しています。	Dim 命令で定義した配列変数の要素数を越えて配列変数を参照しているときに発生します。参照する要素番号を要素数以下に修正するか、定義する要素数を修正してください。
11	Division by Zero	0 による除算を行っています。	0 による除算はできません。0 以外の数値で除算してください。

番号	エラーメッセージ	エラー内容	エラー内容対策
13	Type mismatch	式の左辺、右辺、関数の引数などにおいて、変数の型が一致していません。	式の左辺と右辺、マクロ関数の引数などで型が一致していないときに発生します。正しい型の変数や配列変数を指定してください。 数値に文字列または文字列に数値を代入しているときに発生します。数値には数値、文字列には文字列を代入してください。
15	String too long	文字列変数に代入する文字列が 255 文字を超えています。	255 文字を超える文字列変数は、ふたつの変数に分けて使用してください。
18	Undefined array	定義していない配列を使用しています。	配列変数は使用する前に Dim 命令で定義してください。
23	Line buffer overflow	1 行で入力できる文字の範囲（255 バイト）を超えて入力しようとしています。	おもにシリアル通信やメモリカードから 1 行分のデータを受信するときに発生します。Input$関数を使用して、必要なバイト数ずつ入力してください。
26	FOR without NEXT	Next がないのに for が使用されています。	For Next 文で Next 命令が存在しないか正しく記述されていないときに発生します。Next 命令を正しく記述してください。
32	Undefined label	定義されていないラベルを参照しようとしています。	参照元と参照先のラベル名が異なるときに発生します。定義されているラベル名を使用してください。
121	CASE without SELECT	Select がないのに Case が使用されています。	Select Case 文で Select 命令が存在しないか正しく記述されていないときに発生します。Select 命令を正しく記述してください。
122	END SELECT without SELECT	Select がないのに End Select が使用されています。	
123	SELECT without END SELECT	End Select がないのに Select が使用されています。	Select Case 文で End Select 命令が存在しないか正しく記述されていないときに発生します。End Select 命令を正しく記述してください。
124	CASE without END SELECT	End Select がないのに Case が使用されています。	
125	ELSEIF without IF	If がないのに Elself が使用されています。	If Elself 文で If 命令が存在しないか正しく記述されていないときに発生します。If 命令を正しく記述してください。
126	ELSE without IF	If がないのに Else が使用されています。	
127	ENDIF without IF	If がないのに EndIf が使用されています。	If EndIf 文で If 命令が存在しないか正しく記述されていないときに発生します。If 命令を正しく記述してください。
128	IF without ENDIF	EndIf がないのに If が使用されています。	If EndIf 文で EndIf 命令が存在しないか正しく記述されていないときに発生します。EndIf 命令を正しく記述してください。
129	ELSEIF without ENDIF	EndIf がないのに Elself が使用されています。	
130	ELSE without ENDIF	EndIf がないのに Else が使用されています。	
135	DO without LOOP	Loop がないのに Do が使用されています。	Do Loop While 文で Loop 命令が存在しないか正しく記述されていないときに発生します。Loop 命令を正しく記述してください。
136	LOOP without DO	Do がないのに Loop が使用されています。	Do Loop While 文で Do 命令が存在しないか正しく記述されていないときに発生します。Do 命令を正しく記述してください。
140	EXIT without FOR	For がないのに Exit Do が使用されています。	For Next 文で For 命令が存在しないか正しく記述されていないときに発生します。For 命令を正しく記述してください。

番号	エラーメッセージ	エラー内容	エラー内容対策
141	EXIT without DO	Do がないのに Exit Do が使用されています。	Do Loop While 文で Do 命令が存在しないか正しく記述されていないときに発生します。Do 命令を正しく記述してください。

9-2 予約一覧

マクロ関数、演算子など、システム側であらかじめ定義された文字列を予約語と呼びます。予約語は、マクロカスタマイズ機能のプログラムに使用しないでください。

9-2-1 予約語一覧

FH シリーズで定義されている予約語は以下のとおりです。

A

Abs	AbsolutePath$	Active&
ActiveTable&	AddGlobalData	AddSystemData
And	Append	ApplicationPath$
ApplicationVersion$	ApproximationCircle	ArgmentCountMax&
ArgmentsLength&	ArgmentsLine$	ArgmentString$
ArgmentValue#	ArgumentsLength&	ArgumentString$
ArgumentValue#	ArrayDims	ArrayLen
As	Asc	AssignUnit
Atn		

B

BusyOut	BitPattern&	

C

Calculate?*1	Call	Case
Catch	ChangeScene	ChangeSceneGroup
CheckUnit	Chr$	Clear
ClearMeasureData	ClearScene	ClearSceneGroup
Close	CloseTextData	Cls
CMD$	CmdNo&	CommandArea&
CommandCode&	CommandCountMax&	CommandData&
CommandDelimiter$	CommandDelimiter1$	CommandExecute&
CommandLine$	CommandMemoryAddress&	CommandResponse&
CommandString$	COMMENTSTRING$	Cont
ContinuousMeasure	CopyMeasureImage	CopyScene
CopySceneGroup	CopyUnit	CopyUnitFigure
CopyUnitImage	CopyUnitModel	Cos
CR	CR$	CR_LF$
CreateDialogData	Crspoint	

*1. ?には数字が入ります。

D

DATACOUNT&	Date$	DBL_MAX
DBL_MIN	Debug	DebugPrint
Delete	DeleteDialogData	DeleteProfileData
DeleteUnit	DelimiterCount&	Delimiter1Count&
Dim	DisableUnit	DisplaySubNo
DISPLAYTEXT$	DisplayUnitNo	DISPNAMESTRING$
Do	Dposline	DrawArc

DrawArcW	DrawBox	DrawCircle
DrawCircleW	DrawCursor	DrawEllipse
DrawFigure	DrawFillImage	DrawJudgeText
DrawLine	DrawLineW	DrawMeasureImage
DrawPoint	DrawPolygon	DrawSearchFigure
DrawText	DrawTextG	DrawUnitImage
Dskf		

E

ElapsedTime	Else	Elseif
End	EndIf	Eof
Erase	Erl	Err
ERRCMND$	Errno	Error
ErrorOut	Executable	ExecuteErrorProc
ExecuteImageLogging	Exit	ExitFzProcess
Exp	Explicit	

F

False	Fcopy	FieldbusCommand
FigureType	Files	Fix
FONTSIZE_NORMAL	FONTSTYLE_BOLD	FONTSTYLE_ITALIC
FONTSTYLE_NORMAL	FONTSTYLE_STRIKEOUT	FONTSTYLE_UNDERLINE
For	FunctionCall	

G

GetAll	GetCommandNo	GetDelimiter1Count
GetDelimiterCount	GetGlobalData	GetDialogData
GetImageSize	GetImageWindow	GetMeasureOut
GetPlcData	GetPollingState	GetPort
GetProfileData	GetSceneData	GetSceneGroupData
GetSystemData	GetText$	GetTextWindow
GetUnitData	GetUnitFigure	Global
Gosub	Goto	

H

Hex$	HighBytes&	

I

If	ImageFormat	ImageUpdate
Include	Initialize	InitSetting
Input	Input$	InsertUnit
Int	INT_MAX	INT_MIN
Ioident$	IsFile	IsEndFlg&
ItemCount	Itemident$	ItemInfo
ItemTitle$		

J

JGINDEX&	JUDGE&	JUDGE_ACTIVATEERROR
JUDGE_ERROR	JUDGE_IMAGEERROR	JUDGE_MEMORYERROR
JUDGE_MODELERROR	JUDGE_NC	JUDGE_NG

JUDGE_OK	JUDGELOWER#	JUDGEMACROFLAG&
JudgeOut	JUDGEUPPER#	

K

Keyword	Kill	

L

LanguageString$	LCase$	Left$
Len	LF	LF$
Line	List	Load
LoadBackupData	LoadScene	LoadSceneGroup
LoadSystemData	LoadUnitData	Local
Log	Loop	LowBytes&
Lsqumeth		

M

M_PI	MacroCall	MacroClose
MacroDirect	MacroGosub	MacroOpen
MCRINIT	Measure	MeasureDispG
MeasureDispI	MeasureDispT	MeasureId$
MeasureInit	MeasureOut	MeasureProc
MeasureStart	MeasureStop	MeasureStopTable&
MemoryInfo	MessageBox	Mid$
Mkdir	Mod	MoveUnit

N

New	Next	Not

O

On	Open	OpenTextData
Option	Or	Output

P

ParallelCommand	ParallelExecute	ParaOffset&
Piece$	PlcLinkCommand	PlcRcvData&
PlcRcvStrData$	PlcSndData&	Print
PS_DASH	PS_DASHDOT	PS_DASHDOTDOT
PS_DOT	PS_INSIDEFRAME	PS_NULL
PS_SOLID	PutAll	PutPort

R

RaiseErrorProcEvent	RaiseOptionEvent	ReadPlcMemory
ReceiveData	ReDim	RefreshImageWindow
RefreshJudgeWindow	RefreshTextWindow	RefreshTimeWindow
Rem	Remeasure	RENUMPROC
RenumUnitNo	ResponseArea&	ResponseCode&
ResponseMemoryAddress&	ResponseString$	ResponseValue&
RESULTDATA#	RESULTJUDGE&	Return
RGB	Right$	Rmdir

Run	RunOut	

S

Save	SaveBackupData	SaveData
SaveImage	SaveMeasureImage	SaveScene
SaveSceneGroup	SaveSystemData	SaveUnitData
SceneCount	SceneDescription$	SceneGroupCount
SceneGroupNo	SceneGroupTitle$	SceneMaker$
SceneNo	SceneTitle$	ScreenCapture
Select	SendData	SendString
SetArgument	SetDialogData	SetDisplayUnitNo
SetDrawStyle	SetForegroundLine	SetGlobalData
SetImageWindow	SETJUDGE?*1	SETJUDGESHARED
SetJudgeWindow	SetMeasureImage	SetMeasureOut
SetPlcData	SetPollingState	SetProfileData
SetSceneData	SetSceneDescription	SetSceneGroupData
SetSceneGroupTitle	SetSceneMaker	SetSceneTitle
SetStop	SetSystemData	SetTextStyle
SetTextWindow	SetTimeWindow	SetUnitData
SetUnitFigure	SetUnitJudge	SetUnitTitle
SetupDisp	SetUserSubroutine	SetVar
Shell	Sin	Sqr
StartTimer	Step	Stop
Str$	Str2$	StrCount&
StringCommand	Stupproc	Sub
SubArgmentsLine$	SubList	SystemInfo
SystemReset		

*1. ?には数字が入ります。

T

TA_BASELINE	TA_BOTTOM	TA_CENTER
TA_LEFT	TA_NOUPDATECP	TA_RIGHT
TA_RTLREADING	TA_TOP	TA_UPDATECP
TAB	TAB$	Tan
Task	TEMPLATEVERSION&	TestMeasure
Then	Time$	Timer
TJGFLAG&	TmpSubArgmentsLine$	TMPSTR$
TMPSTROLD$	To	TotalJudge
TransformAngle	TransformArea	TransformDist
TransformLine	TransformXY	True
Try		

U

UCase$	UnitCount	UnitData
UnitData$	UnitData2	UnitInfo
UnitItemIdent$	UnitJudge	UnitNo
UnitTitle$	UserDialog	Ut

V

Val	VarList	VarPop

VarPtr	VarPush	VarSave

W

Wait	While	WritePlcMemory

X

Xor		

9-2-2 グローバルデータの予約語一覧

FH シリーズで予約されているグローバルデータは以下のとおりです。

使用上の注意

センサコントローラの起動時、マクロからグローバルデータは取得できません。

ControlFlowParallelCommand_status	editScnGroupNo	editScnNo
editState	mageWindowOrigin_IoCommand	ImageWindowSubNoMax_IoCommand
LayoutNoLocal_IoCommand	LayoutNoRemote_IoCommand	LineNoLocal_IoCommand
LineNoRemote_IoCommand	ParallelDIOffset	RemoteOperationStatus_IoCommand

9-3 システムデーター覧

システムデータを設定または取得するときに必要な識別情報と識別名は以下のとおりです。

使用上の注意

マニュアルに記載のないデータを変更しないでください。変更すると正常に動作しない場合があります。

識別情報 0	識別情報 1	データ値
Configuration （一般）	language	言語 表示言語を設定します chs：簡体字中国語、cht：繁体字中国語、 deu：ドイツ語、eng：英語、esp：スペイン語、 fra：フランス語、ita：イタリア語、jpn：日本語、 kor：韓国語、pol：ポーランド語、 vie：ベトナム語
	initialSceneGroupNo	起動シーングループ番号
	initialSceneNo	起動シーン番号
	initialMeasureOut	外部に出力する 0：しない 1：する
	operationPriority	操作優先度 0：計測結果表示処理優先 1：メニュー操作優先
	measureInitPriority	計測初期化処理優先度 0：計測トリガ受付を優先 1：画面再描画処理を優先
	measureProcPriority	計測処理優先度 0：計測処理を最高の優先度で実行 1：計測処理の優先度を下げる
	operatingMode	動作モード 0：標準 1：倍速マルチインプット 2：複数ラインランダムトリガ 3：ノンストップ調整
	parallelExecute	並列処理 0：OFF 1：ON
	sceneCount	シーン数
	sceneGroupCount	シーングループ数
	unitDataAllocator	メモリ管理方式 0：使用メモリ最大サイズの規定ありで、解放時のメモリ再配置あり 1：使用メモリ最大サイズの規定ありで、解放時のメモリ再配置なし 2：使用メモリ最大サイズの規定なしで、解放時のメモリ再配置なし
	imageDataAllocator	

識別情報 0	識別情報 1	データ値
ErrorProc （エラー処理）	errorKind00000	システム異常の場合のエラー動作設定 "n1,n2"の文字列で設定します。 • n1：エラー出力 0：外部にエラー出力しない 1：外部にエラー出力する • n2：UI 表示方法 0：エラーを表示しない 1：情報表示ウィンドウ内にエラー内容の通知を行う（ただし「ロギングエラー」、「PLC リンク通信エラー」以外では、エラーダイアログを表示します。） 2：エラーダイアログを表示する 3：エラーダイアログを表示し、ダイアログを閉じるとエラークリアする
	errorKind00001	システム異常（ファン／電圧異常）の場合のエラー動作設定 "n1,n2"の文字列で設定します。 設定内容は、errorKind00000 のデータ値と同じです。 errorKind00000 のデータ値を参照してください。
	errorKind00003	バッテリ異常の場合のエラー動作設定 "n1,n2"の文字列で設定します。 設定内容は、errorKind00000 のデータ値と同じです。 errorKind00000 のデータ値を参照してください。
	errorKind00004	UWF オーバレイ枯渇の場合のエラー動作設定 "n1,n2"の文字列で設定します。 設定内容は、errorKind00000 のデータ値と同じです。 errorKind00000 のデータ値を参照してください。
	errorKind00010	カメラ接続異常の場合のエラー動作設定 "n1,n2"の文字列で設定します。 設定内容は、errorKind00000 のデータ値と同じです。 errorKind00000 のデータ値を参照してください。
	errorKind00011	接続カメラに変更ありの場合のエラー動作設定 "n1,n2"の文字列で設定します。 設定内容は、errorKind00000 のデータ値と同じです。 errorKind00000 のデータ値を参照してください。
	errorKind00012	カメラ過電流検出の場合のエラー動作設定 "n1,n2"の文字列で設定します。 設定内容は、errorKind00000 のデータ値と同じです。 errorKind00000 のデータ値を参照してください。
	errorKind00013	照明機器接続構成異常の場合のエラー動作設定 "n1,n2"の文字列で設定します。 設定内容は、errorKind00000 のデータ値と同じです。 errorKind00000 のデータ値を参照してください。
	errorKind00020	画像ロギングディスク書込異常の場合のエラー動作設定 "n1,n2"の文字列で設定します。 設定内容は、errorKind00000 のデータ値と同じです。 errorKind00000 のデータ値を参照してください。
	errorKind00030	パラレル出力タイムアウトの場合のエラー動作設定 "n1,n2"の文字列で設定します。 設定内容は、errorKind00000 のデータ値と同じです。 errorKind00000 のデータ値を参照してください。
	errorKind00031	PLC リンク通信エラーの場合のエラー動作設定 "n1,n2"の文字列で設定します。 設定内容は、errorKind00000 のデータ値と同じです。 errorKind00000 のデータ値を参照してください。
	errorKind00032	パラレル I/O 過電流検出の場合のエラー動作設定 "n1,n2"の文字列で設定します。 設定内容は、errorKind00000 のデータ値と同じです。 errorKind00000 のデータ値を参照してください。

識別情報 0	識別情報 1	データ値
	errorKind00040	データロード異常の場合のエラー動作設定 "n1,n2"の文字列で設定します。 設定内容は、errorKind00000 のデータ値と同じです。 errorKind00000 のデータ値を参照してください。
	errorKind00041	データ転送異常の場合のエラー動作設定 "n1,n2"の文字列で設定します。 設定内容は、errorKind00000 のデータ値と同じです。 errorKind00000 のデータ値を参照してください。
	errorKind00042	起動時シーングループ番号不正の場合のエラー動作設定 "n1,n2"の文字列で設定します。 設定内容は、errorKind00000 のデータ値と同じです。 errorKind00000 のデータ値を参照してください。
	errorKind00043	起動時シーン番号不正の場合のエラー動作設定 "n1,n2"の文字列で設定します。 設定内容は、errorKind00000 のデータ値と同じです。 errorKind00000 のデータ値を参照してください。
IoModule （通信モジュール）	ioIdent2	シリアル(Ethernet) UdpNormal：無手順(UCP) TcpNormal：無手順(TCP) TcpClient：無手順（TCP クライアント） UdpNormal2：無手順(UDP)（Fxxx シリーズ方式） UdpPlcLink：PLC リンク(SYSMAC CS/CJ/CP/One)(UDP) TcpPlcLink：PLC リンク(SYSMAC CS/CJ/CP/One)(TCP) UdpPlcLinkM：PLC リンク(MELSEC QnU/Q/QnAS)(UDP) TcpPlcLinkM：PLC リンク(MELSEC QnU/Q/QnAS)(TCP) UdpPlcLinkY：PLC リンク(JEPMC MP)
	ioIdent1	シリアル(RS-232C/422) SerialNormal：無手順 SerialNormal2：無手順（Fxxx シリーズ方式） SerialPlcLink：PLC リンク(SYSMAC CS/CJ/CP/One) SerialPlcLinkM：PLC リンク（MELSEC QnU/Q/QnAS シリーズ）
	ioIdent0	パラレル ParallelIo：標準パラレル I/O
	ioIdent3	Fieldbus 空文字：なし EtherCAT：EtherCAT EtherNetIP：EtherNet/IP
	ioIdent4	遠隔操作 RemoteServer：あり 空文字：なし
MultiLineRandom （複数ラインランダム設定）	lineCount	ライン数（2～8）
	physicalCameraMask0	カメラ - ライン割付 0
	physicalCameraMask1	カメラ - ライン割付 1
	physicalCameraMask2	カメラ - ライン割付 2
	physicalCameraMask3	カメラ - ライン割付 3
	physicalCameraMask4	カメラ - ライン割付 4
	physicalCameraMask5	カメラ - ライン割付 5
	physicalCameraMask6	カメラ - ライン割付 6
	physicalCameraMask7	カメラ - ライン割付 7

識別情報 0	識別情報 1	データ値
CameraControl （カメラ設定）	cameraDelay0	カメラ 0 露光開始加算時間（Ver.6.55 以前 STEP - カメラ間ディレイ）
	cameraDelay1	カメラ 1 露光開始加算時間（Ver.6.55 以前 STEP - カメラ間ディレイ）
	cameraDelay2	カメラ 2 露光開始加算時間（Ver.6.55 以前 STEP - カメラ間ディレイ）
	cameraDelay3	カメラ 3 露光開始加算時間（Ver.6.55 以前 STEP - カメラ間ディレイ）
	cameraDelay4	カメラ 4 露光開始加算時間（Ver.6.55 以前 STEP - カメラ間ディレイ）
	cameraDelay5	カメラ 5 露光開始加算時間（Ver.6.55 以前 STEP - カメラ間ディレイ）
	cameraDelay6	カメラ 6 露光開始加算時間（Ver.6.55 以前 STEP - カメラ間ディレイ）
	cameraDelay7	カメラ 7 露光開始加算時間（Ver.6.55 以前 STEP - カメラ間ディレイ）
	transferRate0	カメラ 0 通信速度 0：標準 2：高速
	transferRate1	カメラ 1 通信速度 0：標準 2：高速
	transferRate2	カメラ 2 通信速度 0：標準 2：高速
	transferRate3	カメラ 3 通信速度 0：標準 2：高速
	transferRate4	カメラ 4 通信速度 0：標準 2：高速
	transferRate5	カメラ 5 通信速度 0：標準 2：高速
	transferRate6	カメラ 6 通信速度 0：標準 2：高速
	transferRate7	カメラ 7 通信速度 0：標準 2：高速
	timingSignal	出力信号設定 0：STGOUT 1：SHTOUT
	stgoutCamera	カメラが未接続の STGOUT 信号を出力します 0：出力しない 1：出力する
	shtoutDelay	SHTOUT 信号出力加算時間（Ver.6.55 以前 SHTOUT 信号ディレイ）
	shtoutWidth	SHTOUT 信号パルス幅
	shtoutPolarity	SHTOUT 信号パルス極性 0：負極性 1：正極性

識別情報 0	識別情報 1	データ値
CameraInfo （カメラ情報）	cameraModel0	接続されているカメラの種類
	cameraModel1	
	cameraModel2	
	cameraModel3	
	cameraModel4	
	cameraModel5	
	cameraModel6	
	cameraModel7	
ParallelIo （パラレル）	polarity	出力極性 0：NG 時 ON 1：OK 時 ON
	handshake	出力制御 0：なし 1：ハンドシェイク 2：同期出力
	cycleTime	出力周期
	riseTime	立上時間
	outputTime	出力時間
	timeout	タイムアウト
	delayCount	ディレイ段数
	orOutMode	OR 信号のワンショット出力 0：OFF 1：ON
	orOutputTime	OR 信号のワンショット出力時間
SerialNormal （無手順）	timeout	タイムアウト
	rsMode	通信種別 0：RS-232C
	baudRate	通信速度[bps] (2400/4800/9600/19200/38400/57600/115200)
	byteSize	データ長[bit] (7 or 8)
	parity	パリティ 0：なし 1：奇数 2：偶数
	stopBits	ストップビット[bit] 0：1 1：2
	softFlow	フロー制御 0：なし 1：あり（Xon/Xoff）
	delimiter	デリミタ 0：CR 1：LF 2：CR+LF
SerialNormal2 （無手順（Fxxx シリーズ方式）	timeout	タイムアウト
	rsMode	通信種別 0：RS-232C
	baudRate	通信速度[bps] (2400/4800/9600/19200/38400/57600/115200)
	byteSize	データ長[bit] (7 or 8)
	parity	パリティ 0：なし 1：奇数 2：偶数

識別情報 0	識別情報 1	データ値
	stopBits	ストップビット[bit] 0：1 1：2
	softFlow	フロー制御 0：なし 1：あり（Xon/Xoff）
	delimiter	デリミタ 0：CR 1：LF 2：CR+LF
SerialPlcLink（RS-232C/422 PLC リンク（SYSMAC CS/CJ/CP/One））	handshake	出力制御 0：なし 1：ハンドシェイク
	timeout	タイムアウト
	rsMode	通信種別 0：RS-232C
	baudRate	通信速度[bps] (2400/4800/9600/19200/38400/57600/115200)
	byteSize	データ長[bit] (7 or 8)
	parity	パリティ 0：なし 1：奇数 2：偶数
	stopBits	ストップビット[bit] 0：1 1：2
	softFlow	フロー制御 0：なし 1：あり（Xon/Xoff）
	commandArea	命令エリア エリア種別
	commandMemoryAddress	命令エリア アドレス
	responseArea	応答エリア エリア種別
	responseMemoryAddress	応答エリア アドレス
	outputArea	データ出力エリア エリア種別
	outputMemoryAddress	データ出力エリア アドレス
	outputBuffering	非同期出力 0：OFF 1：ON
	responseTimeout	リトライ間隔
	pollingMinCycle	ポーリング周期
SerialPlcLinkM（RS-232C/422 PLC リンク（MELSEC QnU/Q/QnAS））	handshake	出力制御 0：なし 1：ハンドシェイク
	timeout	タイムアウト
	rsMode	通信種別 0：RS-232C
	baudRate	通信速度[bps] (2400/4800/9600/19200/38400/57600/115200)
	byteSize	データ長[bit] (7 or 8)
	parity	パリティ 0：なし 1：奇数 2：偶数

識別情報 0	識別情報 1	データ値
	stopBits	ストップビット[bit] 0：1 1：2
	softFlow	フロー制御 0：なし 1：あり（Xon/Xoff）
	commandArea	命令エリア エリア種別
	commandMemoryAddress	命令エリア アドレス
	responseArea	応答エリア エリア種別
	responseMemoryAddress	応答エリア アドレス
	outputArea	データ出力エリア エリア種別
	outputMemoryAddress	データ出力エリア アドレス
	outputBuffering	非同期出力 0：OFF 1：ON
	responseTimeout	リトライ間隔
	pollingMinCycle	ポーリング周期
UdpPlcLink（Ethernet PLC リンク(SYSMAC CS/CJ/CP/One)(UDP)）	handshake	出力制御 0：なし 1：ハンドシェイク
	commandArea	命令エリア エリア種別
	commandMemoryAddress	命令エリア アドレス
	responseArea	応答エリア エリア種別
	responseMemoryAddress	応答エリア アドレス
	outputArea	データ出力エリア エリア種別
	outputMemoryAddress	データ出力エリア アドレス
	outputBuffering	非同期出力 0：OFF 1：ON
	responseTimeout	リトライ間隔
	responseTimeout2	リトライ間隔 2
	pollingMinCycle	ポーリング周期
	enableDhcp	自動設定 0：OFF 1：ON
	ipAddress	IP アドレス
	subnetMask	サブネットマスク
	defaultGateway	デフォルトゲートウェイ
	enableDhcp2	自動設定 2 0：OFF 1：ON
	ipAddress2	IP アドレス 2
	subnetMask2	サブネットマスク 2
	defaultGateway2	デフォルトゲートウェイ 2
	dns	DNS サーバ
	dns2	DNS サーバ 2
	wins	WINS サーバ
	wins2	WINS サーバ 2
	destIpAddress	出力 IP アドレス
	portNo	ポート番号

識別情報 0	識別情報 1	データ値
UdpPlcLinkM (Ethernet PLC リンク(MELSEC QnU/Q/QnAS) (UDP)	handshake	出力制御 0：なし 1：ハンドシェイク
	commandArea	命令エリア エリア種別
	commandMemoryAddress	命令エリア アドレス
	responseArea	応答エリア エリア種別
	responseMemoryAddress	応答エリア アドレス
	outputArea	データ出力エリア エリア種別
	outputMemoryAddress	データ出力エリア アドレス
	outputBuffering	非同期出力 0：OFF 1：ON
	responseTimeout	リトライ間隔
	responseTimeout2	リトライ間隔 2
	pollingMinCycle	ポーリング周期
	enableDhcp	自動設定 0：OFF 1：ON
	ipAddress	IP アドレス
	subnetMask	サブネットマスク
	defaultGateway	デフォルトゲートウェイ
	enableDhcp2	自動設定 2 0：OFF 1：ON
	ipAddress2	IP アドレス 2
	subnetMask2	サブネットマスク 2
	defaultGateway2	デフォルトゲートウェイ 2
	dns	DNS サーバ
	dns2	DNS サーバ 2
	wins	WINS サーバ
	wins2	WINS サーバ 2
	destIpAddress	出力 IP アドレス
	portNo	ポート番号
TcpPlcLink (Ethernet PLC リンク(SYSMAC CS/CJ/CP/One) (TCP))	handshake	出力制御 0：なし 1：ハンドシェイク
	commandArea	命令エリア エリア種別
	commandMemoryAddress	命令エリア アドレス
	responseArea	応答エリア エリア種別
	responseMemoryAddress	応答エリア アドレス
	outputArea	データ出力エリア エリア種別
	outputMemoryAddress	データ出力エリア アドレス
	outputBuffering	非同期出力 0：OFF 1：ON
	responseTimeout	リトライ間隔
	responseTimeout2	リトライ間隔 2
	pollingMinCycle	ポーリング周期
	enableDhcp	自動設定 0：OFF 1：ON
	ipAddress	IP アドレス
	subnetMask	サブネットマスク

識別情報 0	識別情報 1	データ値
	defaultGateway	デフォルトゲートウェイ
	enableDhcp2	自動設定 2 0：OFF 1：ON
	ipAddress2	IP アドレス 2
	subnetMask2	サブネットマスク 2
	defaultGateway2	デフォルトゲートウェイ 2
	dns	DNS サーバ
	dns2	DNS サーバ 2
	wins	WINS サーバ
	wins2	WINS サーバ 2
	serverIpAddress	接続先 IP アドレス
	portNo	ポート番号
TcpPlcLinkM (Ethernet PLC リンク(MELSEC QnU/Q/QnAS)(TCP))	handshake	出力制御 0：なし 1：ハンドシェイク
	commandArea	命令エリア エリア種別
	commandMemoryAddress	命令エリア アドレス
	responseArea	応答エリア エリア種別
	responseMemoryAddress	応答エリア アドレス
	outputArea	データ出力エリア エリア種別
	outputMemoryAddress	データ出力エリア アドレス
	outputBuffering	非同期出力 0：OFF 1：ON
	responseTimeout	リトライ間隔
	responseTimeout2	リトライ間隔 2
	pollingMinCycle	ポーリング周期
	enableDhcp	自動設定 0：OFF 1：ON
	ipAddress	IP アドレス
	subnetMask	サブネットマスク
	defaultGateway	デフォルトゲートウェイ
	enableDhcp2	自動設定 2 0：OFF 1：ON
	ipAddress2	IP アドレス 2
	subnetMask2	サブネットマスク 2
	defaultGateway2	デフォルトゲートウェイ 2
	dns	DNS サーバ
	dns2	DNS サーバ 2
	wins	WINS サーバ
	wins2	WINS サーバ 2
	serverIpAddress	接続先 IP アドレス
	portNo	ポート番号
UdpNormal (無手順(UDP))	enableDhcp	自動設定 0：OFF 1：ON
	ipAddress	IP アドレス
	subnetMask	サブネットマスク
	defaultGateway	デフォルトゲートウェイ

識別情報 0	識別情報 1	データ値
	enableDhcp2	自動設定 2 0：OFF 1：ON
	ipAddress2	IP アドレス 2
	subnetMask2	サブネットマスク 2
	defaultGateway2	デフォルトゲートウェイ 2
	dns	DNS サーバ
	dns2	DNS サーバ 2
	wins	WINS サーバ
	wins2	WINS サーバ 2
	destIpAddress	出力 IP アドレス
	portNo	入力ポート番号
	portNo2	出力ポート番号
TcpNormal (無手順(TCP))	enableDhcp	自動設定 0：OFF 1：ON
	ipAddress	IP アドレス
	subnetMask	サブネットマスク
	defaultGateway	デフォルトゲートウェイ
	enableDhcp2	自動設定 2 0：OFF 1：ON
	ipAddress2	IP アドレス 2
	subnetMask2	サブネットマスク 2
	defaultGateway2	デフォルトゲートウェイ 2
	dns	DNS サーバ
	dns2	DNS サーバ 2
	wins	WINS サーバ
	wins2	WINS サーバ 2
	portNo	ポート番号
TcpClient (無手順(TCP Client))	enableDhcp	自動設定 0：OFF 1：ON
	ipAddress	IP アドレス
	subnetMask	サブネットマスク
	defaultGateway	デフォルトゲートウェイ
	enableDhcp2	自動設定 2 0：OFF 1：ON
	ipAddress2	IP アドレス 2
	subnetMask2	サブネットマスク 2
	defaultGateway2	デフォルトゲートウェイ 2
	dns	DNS サーバ
	dns2	DNS サーバ 2
	wins	WINS サーバ
	wins2	WINS サーバ 2
	portNo	ポート番号
	serverIpAddress	接続先 IP アドレス
UdpNormal2 (無手順 (UDP Fxxx シリーズ方式))	enableDhcp	自動設定 0：OFF 1：ON
	ipAddress	IP アドレス
	subnetMask	サブネットマスク

識別情報 0	識別情報 1	データ値
	defaultGateway	デフォルトゲートウェイ
	enableDhcp2	自動設定 2 0：OFF 1：ON
	ipAddress2	IP アドレス 2
	subnetMask2	サブネットマスク 2
	defaultGateway2	デフォルトゲートウェイ 2
	dns	DNS サーバ
	dns2	DNS サーバ 2
	wins	WINS サーバ
	wins2	WINS サーバ 2
	destIpAddress	出力 IP アドレス
	portNo	入力ポート番号
	portNo2	出力ポート番号
UdpPlcLinkY (Ethernet PLC リンク(JEPMC MP))	handshake	出力制御 0：なし 1：ハンドシェイク
	commandArea	命令エリア エリア種別
	commandMemoryAddress	命令エリア アドレス
	responseArea	応答エリア エリア種別
	responseMemoryAddress	応答エリア アドレス
	outputArea	データ出力エリア エリア種別
	outputMemoryAddress	データ出力エリア アドレス
	outputBuffering	非同期出力 0：OFF 1：ON
	responseTimeout	リトライ間隔
	responseTimeout2	リトライ間隔 2
	pollingMinCycle	ポーリング周期
	enableDhcp	自動設定 0：OFF 1：ON
	ipAddress	IP アドレス
	subnetMask	サブネットマスク
	defaultGateway	デフォルトゲートウェイ
	enableDhcp2	自動設定 2 0：OFF 1：ON
	ipAddress2	IP アドレス 2
	subnetMask2	サブネットマスク 2
	defaultGateway2	デフォルトゲートウェイ 2
	dns	DNS サーバ
	dns2	DNS サーバ 2
	wins	WINS サーバ
	wins2	WINS サーバ 2
	portNo	ポート番号
	destIpAddress	出力 IP アドレス
EtherCAT (EtherCAT)	handshake	出力制御 0：なし 1：ハンドシェイク
	cycleTime	出力周期
	outputTime	出力時間
	timeout	タイムアウト

識別情報 0	識別情報 1	データ値
	outputDataSize0	ライン 0 データ出力個数 32：Result Data Format 0(DINT 8 個) 64：Result Data Format 1(DINT 16 個) 128：Result Data Format 2(DINT 32 個) 256：Result Data Format 3(DINT 64 個) 2097152：Result Data Format 4(LREAL 4 個) 4194304：Result Data Format 5(LREAL 8 個) 8388608：Result Data Format 6(LREAL 16 個) 16777216：Result Data Format 7(LREAL 32 個) 1572872：Result Data Format 8(DINT 2 + LREAL 3 個) 3145744：Result Data Format 9(DINT 4 + LREAL 6 個) 6291488：Result Data Format 10(DINT 8 + LREAL 12 個) 12582976：Result Data Format 11(DINT 16 + LREAL 24 個)
	outputDataSize1	ライン 1 データ出力個数 設定内容は、outputDataSize0 のデータ値と同じです。 outputDataSize0 のデータ値を参照してください。
	outputDataSize2	ライン 2 データ出力個数 設定内容は、outputDataSize0 のデータ値と同じです。 outputDataSize0 のデータ値を参照してください。
	outputDataSize3	ライン 3 データ出力個数 設定内容は、outputDataSize0 のデータ値と同じです。 outputDataSize0 のデータ値を参照してください。
	outputDataSize4	ライン 4 データ出力個数 設定内容は、outputDataSize0 のデータ値と同じです。 outputDataSize0 のデータ値を参照してください。
	outputDataSize5	ライン 5 データ出力個数 設定内容は、outputDataSize0 のデータ値と同じです。 outputDataSize0 のデータ値を参照してください。
	outputDataSize6	ライン 6 データ出力個数 設定内容は、outputDataSize0 のデータ値と同じです。 outputDataSize0 のデータ値を参照してください。
	outputDataSize7	ライン 7 データ出力個数 設定内容は、outputDataSize0 のデータ値と同じです。 outputDataSize0 のデータ値を参照してください。
	userOutputDataSize0	ライン 0 ユーザ出力データ個数
	userOutputDataSize1	ライン 1 ユーザ出力データ個数
	userOutputDataSize2	ライン 2 ユーザ出力データ個数
	userOutputDataSize3	ライン 3 ユーザ出力データ個数
	userOutputDataSize4	ライン 4 ユーザ出力データ個数
	userOutputDataSize5	ライン 5 ユーザ出力データ個数
	userOutputDataSize6	ライン 6 ユーザ出力データ個数
	userOutputDataSize7	ライン 7 ユーザ出力データ個数
	userInputDataSize0	ライン 0 ユーザ入力データ個数
	userInputDataSize1	ライン 1 ユーザ入力データ個数
	userInputDataSize2	ライン 2 ユーザ入力データ個数
	userInputDataSize3	ライン 3 ユーザ入力データ個数
	userInputDataSize4	ライン 4 ユーザ入力データ個数
	userInputDataSize5	ライン 5 ユーザ入力データ個数
	userInputDataSize6	ライン 6 ユーザ入力データ個数
	userInputDataSize7	ライン 7 ユーザ入力データ個数

識別情報 0	識別情報 1	データ値
EtherNet/IP (EtherNet/IP)	handshake	出力制御 0：なし 1：ハンドシェイク
	cycleTime	出力周期
	outputTime	出力時間
	timeout	タイムアウト
	outputDataSize	出力データサイズ
	userOutputDataSize	ユーザデータエリア出力データサイズ 【0】：なし 32：あり
	userInputDataSize	ユーザデータエリア入力データサイズ 【0】：なし 32：あり
Measure (計測)	fanControl	ファン制御設定 0：低速回転 1：高速回転
	captureDirectory	キャプチャ画像保存先フォルダ
	stepError	計測中入力 0：計測中 STEP 入力時に、ERROR 信号出力 OFF 1：計測中 STEP 入力時に、ERROR 信号出力 ON
	sceneGroupSave	シーングループ切替 0：シーングループ切替時セーブなし 1：シーングループ切替時セーブあり
	sceneWaitTime	シーン切替時間
Trigger (その他)	digitalFilter	STEP 信号フィルタ幅 5：100 11：200 18：300 24：400 30：500
	encoderEnabled	エンコーダトリガを使用する 0：使用しない 1：使用する
	encoderInputPulse	エンコーダ入力選択 0：オープンコレクタ 1：ラインドライバ
	encoderResolutionA	A 相分解能
	encoderZ	1 回転を Z 相入力で判別する 0：判別しない 1：判別する
	encoderStart	信号有効タイミング 0：常に有効 1：Step 入力後に有効
	encoderCount	トリガ信号とリセットタイミング 0：トリガ信号：A 相、リセットタイミング：トリガ毎 1：トリガ信号：A 相、リセットタイミング：回転毎 2：トリガ信号：Z 相、リセットタイミング：トリガ毎
	encoderTriggerAngleA0	A 相トリガパルス 0～5（リセットタイミング：回転毎）
	encoderTriggerAngleA1	
	encoderTriggerAngleA2	
	encoderTriggerAngleA3	
	encoderTriggerAngleA4	
	encoderTriggerAngleA5	

識別情報 0	識別情報 1	データ値
	encoderBacklash	回転方向検知（B 相） 0：検知しない 1：検知する
	encoderBacklashTrigger	逆回転時にトリガを発生させる 0：発生させない 1：発生させる
	encoderTriggerCountA	A 相トリガパルス（リセットタイミング：トリガ毎）
	encoderTriggerCountZ	Z 相トリガパルス
NetworkDrive （ネットワークドライブ設定	drive0	S
	drive1	T
	drive2	U
	drive3	V
	drive4	W
	drive5	X
	drive6	Y
	drive7	Z
	passwordCrypt0	ネットワークドライブの暗号化パスワード
	passwordCrypt1	
	passwordCrypt2	
	passwordCrypt3	
	passwordCrypt4	
	passwordCrypt5	
	passwordCrypt6	
	passwordCrypt7	
Logging （ロギング設定）	imageLogging	画像ロギング 0：保存しない 1：NG の場合保存する 2：すべて保存する
	imageLoggingDirectory	保存先 画像ファイルの保存先（RAMDisk または USB メモリなどの外部ストレージ）を指定します
	imageLoggingHeader	接頭語
	imageLoggingFooter	接尾語
	imageLoggingUseMeasureID	計測 ID を表示する 0：無効 1：有効
	imageLoggingDistributionMode	フォルダ振分 0：手動設定 1：自動設定
	imageLoggingDate	日付ごとに振り分ける 0：無効 1：有効
	imageLoggingScene	シーンごとに振り分ける 0：無効 1：有効 2：有効（フォルダ名にシーン名を使用する）
	imageLoggingJudge	総合判定ごとに振り分ける 0：無効 1：有効
	imageLoggingSubDirectory	サブフォルダ名
	imageLoggingType	複数回撮像画像ロギング single：なし multiple：あり

識別情報 0	識別情報 1	データ値
	imageLoggingFormat	ファイル形式 ifz bfz jfz pfz
	imageLoggingPriority	ロギングモード切替 0：ロギング動作優先 1：計測タクト優先
	dataLogging	データロギング 0：保存しない 1：NG の場合保存する 2：すべて保存する
	dataLoggingDirectory	データロギング保存先フォルダ
FtpImageLogging （FTP クライアント画像ロギング設定）	enabled	本体メモリ+FTP サーバに保存する 0：ロギングしない 1：FTP クライアントでロギングする
	serverIpAddress	接続先 IP アドレス
	userName	ユーザ名
	password	パスワード
	passwordCrypt	FTP サーバの暗号化パスワード
	portNo	ポート番号
	passiveMode	接続方式 0：アクティブ 1：パッシブ
	targetDirectory	フォルダ名
OperationLog （操作ログ設定）	targetDirectory	保存先
	autoEnabled	起動時に開始する 0：開始しない 1：開始する
PanDA （ユーザ定義）	PanDA	-

9-4 IO モジュール一覧

外部装置との通信に必要な I/O モジュールの設定は以下のとおりです。

識別名	IO モジュール名	参照先
EtherCAT	EtherCAT 通信	P.9-26
EtherNetIP	EtherNetIP 通信	P.9-28
PROFINET	PROFINET 通信	P.9-31
ParallelIo	パラレル I/O	P.9-34
SerialNormal SerialNormal2（Fxxx シリーズ方式）	シリアル無手順通信	P.9-36
SerialPlcLinkM	PLC リンク(MELSEC QnU/Q/QnAS)	P.9-38
SerialPlcLink	PLC リンク(SYSMAC CS/CJ/CP/One)	P.9-40
TcpClient	無手順（TCP クライアント）	P.9-43
TcpNormal	無手順(TCP) 通信	P.9-45
UdpNormal UdpNormal2（Fxxx シリーズ方式）	UDP 無手順通信	P.9-47
UdpPlcLinkM	PLC リンク(MELSEC QnU/Q/QnAS)(UDP)	P.9-49
UdpPlcLinkY	PLC リンク(JEPMC MP)	P.9-52
UdpPlcLink	PLC リンク(SYSMAC CS/CJ/CP/One)(UDP)	P.9-55
TcpPlcLinkM	PLC リンク(MELSEC QnU/Q/QnAS)(TCP)	P.9-58
TcpPlcLink	PLC リンク(SYSMAC CS/CJ/CP/One)(TCP)	P.9-61

9-4-1 EtherCAT

EtherCAT 通信

・IoModule 識別名
EtherCAT

・概要
EtherCAT プロトコルを用いて、コマンドおよびデータの送受信を行います。

・システムデータ設定項目

識別名	意味	初期値
handshake	出力制御 0：なし 1：ハンドシェイク	0
cycleTime	出力周期 [EtherCAT 通信周期]	2
outputTime	出力時間 [EtherCAT 通信周期]	1
timeout	タイムアウト[0.1s]	100
outputDataSize0～7	出力エリアデータフォーマット設定	32
userOutputDataSize0～7	ユーザ出力フォーマット設定	0
userInputDataSize0～7	ユーザ入力フォーマット設定	0

・サポートする機能

IoInitialize	OK	-
GetPort	-	-
PutPort	-	-
GetAll	-	-
PutAll	-	-
BusyOut	OK	BUSY の状態を出力します。（参照：「BusyOut (P.8-34)」）
JudgeOut	OK	判定結果を外部へ出力します。（参照：「JudgeOut (P.8-159)」）
RunOut	OK	RUN の状態を出力します。（参照：「RunOut (P.8-225)」）
ReceiveData	-	-
SendData	OK	データを送信します。（参照：「SendData (P.8-246)」）
SendString	-	-
ReadPlcMemory	OK	PLC のメモリ領域から値を読み込みます。（参照：「ReadPlcMemory (P.8-210)」）
WritePlcMemory	OK	PLC のメモリ領域に値を書き込みます。（参照：「WritePlcMemory (P.8-329)」）
SetPlcData	OK	WritePlcMemory 関数で書き込むデータを作成します。（参照：「SetPlcData (P.8-258)」）
GetPlcData	OK	ReadPlcMemory 関数で読み込んだデータを取り出します。（参照：「GetPlcData (P.8-126)」）
SetPollingState	OK	通信モジュールの実行状態を設定します。（参照：「SetPollingState (P.8-260)」）
GetPollingState	OK	通信モジュールの実行状態を取得します。（参照：「GetPollingState (P.8-127)」）
ErrorOut	OK	エラーの状態を出力します。（参照：「ErrorOut (P.8-108)」）

・使用例
SendData を使用してデータを送信します。
SendData のパラメータ、パラメータサイズは不要です。
出力するデータはシステム設定の EtherCAT の出力個数の設定に合わせて使用して下さい。
WritePlcMemory の<channelOffset> は、「8 +（出力データサイズ (byte)）/2」となるように設定してください。

```
Dim DATA&(7)
DATA&(0) = 100
DATA&(1) = 200
DATA&(2) = 300
DATA&(3) = 400
DATA&(4) = 500
DATA&(5) = 600
DATA&(6) = 700
DATA&(7) = 800

' 整数型データ 8 つ分のデータを送信します
SendData "EtherCAT", data&(), 4*8
```

ユーザエリアを使って読み書きができます。

ユーザ入力エリアからの読み込み

```
Dim BUF&(7)
Dim DATA_DINT&(3)
Dim DATA_LREAL#(1)

Rem EtherCAT メモリを読み込みます
ReadPlcMemory "EtherCAT", 0, 10, 16, BUF&()

Rem バッファから DINT 型と LREAL 型のデータを取り出します
GetPlcData "EtherCAT", BUF&(), 0, 4, DATA_DINT&(0)
GetPlcData "EtherCAT", BUF&(), 4, 4, DATA_DINT&(1)
GetPlcData "EtherCAT", BUF&(), 8, 4, DATA_DINT&(2)
GetPlcData "EtherCAT", BUF&(), 12, 4, DATA_DINT&(3)
GetPlcData "EtherCAT", BUF&(), 16, 8, DATA_LREAL#(0)
GetPlcData "EtherCAT", BUF&(), 24, 8, DATA_LREAL#(1)
```

ユーザ出力エリアへの書き込み

```
Dim BUF&(7)
Dim DATA_DINT&(3)
Dim DATA_LREAL#(1)

Rem 書き込みたいデータをセットします
DATA_DINT&(0) = 100
DATA_DINT&(1) = 200
DATA_DINT&(2) = 300
DATA_DINT&(3) = 400
DATA_LREAL#(0) = 12.34
DATA_LREAL#(1) = 56.78

Rem バッファへ DINT 型と LREAL 型のデータをセットします
SetPlcData "EtherCAT", BUF&(), 0, 4, DATA_DINT&(0)
SetPlcData "EtherCAT", BUF&(), 4, 4, DATA_DINT&(1)
SetPlcData "EtherCAT", BUF&(), 8, 4, DATA_DINT&(2)
SetPlcData "EtherCAT", BUF&(), 12, 4, DATA_DINT&(3)
SetPlcData "EtherCAT", BUF&(), 16, 8, DATA_LREAL#(0)
SetPlcData "EtherCAT", BUF&(), 24, 8, DATA_LREAL#(1)

Rem EtherCAT メモリへ書き込みます
WritePlcMemory "EtherCAT", 0, 24, 16, BUF&()
```

9-4-2 EtherNetIP

EtherNetIP 通信

・IoModule 識別名
EtherNetIP

・概要
EtherNetIP プロトコルを用いて、コマンドおよびデータの送受信を行います。

・システムデータ設定項目

識別名	意味	初期値
handshake	出力制御 0：なし 1：ハンドシェイク	0
cycleTime	出力周期[ms]	100
outputTime	出力時間[ms]	50
timeout	タイムアウト[0.1s]	100

・サポートする機能

IoInitialize	OK	-
GetPort	-	-
PutPort	-	-
GetAll	-	-
PutAll	-	-
BusyOut	-	-
JudgeOut	OK	判定結果を外部へ出力します。（参照：「JudgeOut (P.8-159)」）
RunOut	OK	RUN の状態を出力します。（参照：「RunOut (P.8-225)」）
ReceiveData	-	-
SendData	OK	データを送信します。（参照：「SendData (P.8-246)」）
SendString	-	-
ReadPlcMemory	OK	PLC のメモリ領域から値を読み込みます。（参照：「ReadPlcMemory (P.8-210)」）
WritePlcMemory	OK	PLC のメモリ領域に値を書き込みます。（参照：「WritePlcMemory (P.8-329)」）
SetPlcData	OK	WritePlcMemory 関数で書き込むデータを作成します。（参照：「SetPlcData (P.8-258)」）
GetPlcData	OK	ReadPlcMemory 関数で読み込んだデータを取り出します。（参照：「GetPlcData (P.8-126)」）
SetPollingState	OK	通信モジュールの実行状態を設定します。（参照：「SetPollingState (P.8-260)」）
GetPollingState	OK	通信モジュールの実行状態を取得します。（参照：「GetPollingState (P.8-127)」）
ErrorOut	OK	エラーの状態を出力します。（参照：「ErrorOut (P.8-108)」）

・使用例
SendData を使用してデータを送信します。
Ethernet を使用するため、パラメータに IP アドレスとパラメータサイズ（整数型 × 4 領域）を設定します。

```
Dim DATA&(4)
DATA&(0)=111
DATA&(1)=222
DATA&(2)=333
DATA&(3)=444
DATA&(4)=555

' 整数型データ 5 つ分のデータを送信します
SendData "EtherNetIp", data&(), 4*5
```

ReadPlcMemory を使用して命令エリアの先頭から 7ch 分のデータを取得します。

```
Dim DATA&(7)
ReadPlcMemory "EtherNetIP", 0, 0, 16, DATA&()
For I&= 0 To 7
   Print DATA&(I&)
Next
```

WritePlcMemory を使用してデータ出力エリアに 4ch 分のデータを出力します。

```
Dim BUF&(7)
Dim DATA_DINT&(3)

Rem 書き込みたいデータをセットします
DATA_DINT&(0) = 100
DATA_DINT&(1) = 200
DATA_DINT&(2) = 300
DATA_DINT&(3) = 400

Rem バッファへ DINT 型のデータをセットします
SetPlcData "EtherNetIP", BUF&(), 0, 4, DATA_DINT&(0)
SetPlcData "EtherNetIP", BUF&(), 4, 4, DATA_DINT&(1)
SetPlcData "EtherNetIP", BUF&(), 8, 4, DATA_DINT&(2)
SetPlcData "EtherNetIP", BUF&(), 12, 4, DATA_DINT&(3)

WritePlcMemory "EtherNetIP", 0, 8, 16, BUF&()
```

ユーザエリアを使って読み書きができます。

ユーザ入力エリアからの読み込み

```
Dim BUF&(7)
Dim DATA_DINT&(3)
Dim DATA_LREAL#(1)

Rem EtherNet/IP メモリを読み込みます
ReadPlcMemory "EtherNetIP", 0, 10, 16, BUF&()

Rem バッファから DINT 型と LREAL 型のデータを取り出します
GetPlcData "EtherNetIP", BUF&(), 0, 4, DATA_DINT&(0)
GetPlcData "EtherNetIP", BUF&(), 4, 4, DATA_DINT&(1)
GetPlcData "EtherNetIP", BUF&(), 8, 4, DATA_DINT&(2)
GetPlcData "EtherNetIP", BUF&(), 12, 4, DATA_DINT&(3)
GetPlcData "EtherNetIP", BUF&(), 16, 8, DATA_LREAL#(0)
GetPlcData "EtherNetIP", BUF&(), 24, 8, DATA_LREAL#(1)
```

ユーザ出力エリアへの書き込み

```
Dim BUF&(7)
Dim DATA_DINT&(3)
Dim DATA_LREAL#(1)

Rem 書き込みたいデータをセットします
DATA_DINT&(0) = 100
DATA_DINT&(1) = 200
DATA_DINT&(2) = 300
DATA_DINT&(3) = 400
DATA_LREAL#(0) = 12.34
DATA_LREAL#(1) = 56.78

Rem バッファへ DINT 型と LREAL 型のデータをセットします
SetPlcData "EtherNetIP", BUF&(), 0, 4, DATA_DINT&(0)
SetPlcData "EtherNetIP", BUF&(), 4, 4, DATA_DINT&(1)
SetPlcData "EtherNetIP", BUF&(), 8, 4, DATA_DINT&(2)
SetPlcData "EtherNetIP", BUF&(), 12, 4, DATA_DINT&(3)
SetPlcData "EtherNetIP", BUF&(), 16, 8, DATA_LREAL#(0)
SetPlcData "EtherNetIP", BUF&(), 24, 8, DATA_LREAL#(1)

Rem EtherNet/IP メモリへ書き込みます
WritePlcMemory "EtherNetIP", 0, 24, 16, BUF&()
```

9-4-3 PROFINET

PROFINET 通信

・IoModule 識別名
PROFINET

・概要
PROFINET プロトコルを用いて、コマンドおよびデータの送受信を行います。

・システムデータ設定項目

識別名	意味	初期値
handshake	出力制御 0：なし 1：ハンドシェイク	0
cycleTime	出力周期[ms]	100
outputTime	出力時間[ms]	50
timeout	タイムアウト[0.1s]	100

・サポートする機能

IoInitialize	OK	-
GetPort	-	-
PutPort	-	-
GetAll	-	-
PutAll	-	-
BusyOut	-	-
JudgeOut	OK	判定結果を外部へ出力します。（参照：「JudgeOut (P.8-159)」）
RunOut	OK	RUN の状態を出力します。（参照：「RunOut (P.8-225)」）
ReceiveData	-	-
SendData	OK	データを送信します。（参照：「SendData (P.8-246)」）
SendString	-	-
ReadPlcMemory	OK	PLC のメモリ領域から値を読み込みます。（参照：「ReadPlcMemory (P.8-210)」）
WritePlcMemory	OK	PLC のメモリ領域に値を書き込みます。（参照：「WritePlcMemory (P.8-329)」）
SetPlcData	OK	WritePlcMemory 関数で書き込むデータを作成します。（参照：「SetPlcData (P.8-258)」）
GetPlcData	OK	ReadPlcMemory 関数で読み込んだデータを取り出します。（参照：「GetPlcData (P.8-126)」）
SetPollingState	OK	通信モジュールの実行状態を設定します。（参照：「SetPollingState (P.8-260)」）
GetPollingState	OK	通信モジュールの実行状態を取得します。（参照：「GetPollingState (P.8-127)」）
ErrorOut	OK	エラーの状態を出力します。（参照：「ErrorOut (P.8-108)」）

・使用例
SendData を使用してデータを送信します。
Ethernet を使用するため、パラメータに IP アドレスとパラメータサイズ（整数型 × 4 領域）を設定します。

```
Dim DATA&(4)
DATA&(0)=111
DATA&(1)=222
DATA&(2)=333
DATA&(3)=444
DATA&(4)=555

' 整数型データ 5 つ分のデータを送信します
SendData "PROFINET", data&(), 4*5
```

ReadPlcMemory を使用して命令エリアの先頭から 7ch 分のデータを取得します。

```
Dim DATA&(7)
ReadPlcMemory "PROFINET", 0, 0, 16, DATA&()
For I&= 0 To 7
   Print DATA&(I&)
Next
```

WritePlcMemory を使用してデータ出力エリアに 4ch 分のデータを出力します。

```
Dim BUF&(7)
Dim DATA_DINT&(3)

Rem 書き込みたいデータをセットします
DATA_DINT&(0) = 100
DATA_DINT&(1) = 200
DATA_DINT&(2) = 300
DATA_DINT&(3) = 400

Rem バッファへ DINT 型のデータをセットします
SetPlcData "PROFINET", BUF&(), 0, 4, DATA_DINT&(0)
SetPlcData "PROFINET", BUF&(), 4, 4, DATA_DINT&(1)
SetPlcData "PROFINET", BUF&(), 8, 4, DATA_DINT&(2)
SetPlcData "PROFINET", BUF&(), 12, 4, DATA_DINT&(3)

WritePlcMemory "PROFINET", 0, 8, 16, BUF&()
```

ユーザエリアを使って読み書きができます。

ユーザ入力エリアからの読み込み

```
Dim BUF&(7)
Dim DATA_DINT&(3)
Dim DATA_LREAL#(1)

Rem PROFINET メモリを読み込みます
ReadPlcMemory "PROFINET", 0, 10, 16, BUF&()

Rem バッファから DINT 型と LREAL 型のデータを取り出します
GetPlcData "PROFINET", BUF&(), 0, 4, DATA_DINT&(0)
GetPlcData "PROFINET", BUF&(), 4, 4, DATA_DINT&(1)
GetPlcData "PROFINET", BUF&(), 8, 4, DATA_DINT&(2)
GetPlcData "PROFINET", BUF&(), 12, 4, DATA_DINT&(3)
GetPlcData "PROFINET", BUF&(), 16, 8, DATA_LREAL#(0)
GetPlcData "PROFINET", BUF&(), 24, 8, DATA_LREAL#(1)
```

ユーザ出力エリアへの書き込み

```
Dim BUF&(7)
Dim DATA_DINT&(3)
Dim DATA_LREAL#(1)

Rem 書き込みたいデータをセットします
DATA_DINT&(0) = 100
DATA_DINT&(1) = 200
DATA_DINT&(2) = 300
DATA_DINT&(3) = 400
DATA_LREAL#(0) = 12.34
DATA_LREAL#(1) = 56.78

Rem バッファへ DINT 型と LREAL 型のデータをセットします
SetPlcData "PROFINET", BUF&(), 0, 4, DATA_DINT&(0)
SetPlcData "PROFINET", BUF&(), 4, 4, DATA_DINT&(1)
SetPlcData "PROFINET", BUF&(), 8, 4, DATA_DINT&(2)
SetPlcData "PROFINET", BUF&(), 12, 4, DATA_DINT&(3)
SetPlcData "PROFINET", BUF&(), 16, 8, DATA_LREAL#(0)
SetPlcData "PROFINET", BUF&(), 24, 8, DATA_LREAL#(1)

Rem PROFINET メモリへ書き込みます
WritePlcMemory "PROFINET", 0, 24, 16, BUF&()
```

9-4-4 Parallello

パラレル I/O

・IoModule 識別名
Parallello

・概要
パラレルインタフェースを用いて、コマンドおよびデータの送受信を行います。

・システムデータ設定項目

識別名	意味	初期値
polarity	出力極性 0：NG 時 ON 1：OK 時 ON	0
handshake	出力制御 0：なし 1：ハンドシェイク 2：同期出力	0
cycleTime	出力周期[0.1ms]	100
riseTime	立上時間[0.1ms]	10
outputTime	出力時間[0.1ms]	50
timeout	タイムアウト[0.1ms]	100
delayCount	ディレイ段数	1
orOutMode	OR 信号のワンショット出力 0：OFF 1：ON	0
orOutputTime	OR 信号のワンショット出力時間[0.1ms]	50

・サポートする機能

IoInitialize	OK	-
GetPort	OK	指定した入力端子の入力状態を取得します。（参照：「GetPort (P.8-129)」）
PutPort	OK	出力端子に 1 点出力を行います。（参照：「PutPort (P.8-205)」）
GetAll	OK	入力端子の全点入力を行います。（参照：「GetAll (P.8-120)」）
PutAll	OK	出力端子に全点出力を行います。（参照：「PutAll (P.8-204)」）
BusyOut	OK	BUSY の状態を出力します。（参照：「BusyOut (P.8-34)」）
JudgeOut	OK	判定結果を外部へ出力します。（参照：「JudgeOut (P.8-159)」）
RunOut	OK	RUN の状態を出力します。（参照：「RunOut (P.8-225)」）
ReceiveData	-	-
SendData	OK	データを送信します。（参照：「SendData (P.8-246)」）
SendString	-	-
ReadPlcMemory	-	-
WritePlcMemory	-	-
SetPlcData	-	-
GetPlcData	-	-
SetPollingState	OK	通信モジュールの実行状態を設定します。（参照：「SetPollingState (P.8-260)」）
GetPollingState	OK	通信モジュールの実行状態を取得します。（参照：「GetPollingState (P.8-127)」）
ErrorOut	OK	エラーの状態を出力します。（参照：「ErrorOut (P.8-108)」）

・使用例
SendData を使用してデータを送信します。

```
Dim data&(256)

' 整数型データ 5 つ分のデータを送信します
SendData "Parallello", data&(), 4*5
```

9-4-5 SerialNormal

シリアル無手順通信

・IoModule 識別名

SerialNormal
SerialNormal2（Fxxx シリーズ方式）

・概要

シリアルインタフェースを用いて、コマンドおよびデータの送受信を行います。

・システムデータ設定項目

識別名	意味	初期値
rsMode	通信種別 0：RS-232C	0
baudRate	通信速度[bps]	38400
byteSize	データ長[bit] (7 or 8)	8
parity	パリティ 0：なし 1：奇数 2：偶数	0
stopBits	ストップビット[bit] 0：1 1：2	0
softFlow	フロー制御 0：なし 1：あり（Xon/Xoff）	0
delimiter	デリミタ 0：CR 1：LF 2：CR+LF	0
timeout	タイムアウト[s]	5

・サポートする機能

IoInitialize	OK	-
GetPort	-	-
PutPort	-	-
GetAll	-	-
PutAll	-	-
BusyOut	-	-
JudgeOut	-	-
RunOut	-	-
ReceiveData	OK	データを受信します。（参照：「ReceiveData (P.8-212)」）
SendData	OK	データを送信します。（参照：「SendData (P.8-246)」）
SendString	OK	文字列データを送信します。（参照：「SendString (P.8-248)」）
ReadPlcMemory	-	-
WritePlcMemory	-	-
SetPlcData	-	-
GetPlcData	-	-
SetPollingState	OK	通信モジュールの実行状態を設定します。（参照：「SetPollingState (P.8-260)」）
GetPollingState	OK	通信モジュールの実行状態を取得します。（参照：「GetPollingState (P.8-127)」）
ErrorOut	-	-

・使用例

ReceiveData を使用してデータを受信します。
ReceiveData のパラメータ、パラメータサイズは不要です。

```
Dim data&(256)

' 整数型データ 5 つ分のデータを取得します
ReceiveData "SerialNormal", data&(), 4*5, size&
```

SendData を使用してデータを送信します。

```
Dim data&(256)

' 整数型データ 5 つ分のデータを送信します
SendData "SerialNormal", data&(), 4*5
```

9-4-6 SerialPlcLinkM

PLC リンク(MELSEC QnU/Q/QnAS)

・IoModule 識別名
SerialPlcLinkM

・概要
シリアル PLC リンクインタフェースを用いて、コマンドおよびデータの送受信を行います。

・システムデータ設定項目

識別名	意味	初期値
rsMode	通信種別 0：RS-232C	0
baudRate	通信速度[bps]	9600
byteSize	データ長[bit] (7 or 8)	7
parity	パリティ 0：なし 1：奇数 2：偶数	2
stopBits	ストップビット[bit] 0：1 1：2	1
softFlow	フロー制御 0：なし 1：あり（Xon/Xoff）	0
timeout	タイムアウト[s]	5

・PLC リンク設定項目

識別名	意味	初期値
commandArea	命令エリア エリア種別	チャネル IO(CIO)
commandMemoryAddress	命令エリア アドレス	0
responseArea	応答エリア エリア種別	チャネル IO(CIO)
responseMemoryAddress	応答エリア アドレス	100
outputArea	データ出力エリア エリア種別	チャネル IO(CIO)
outputMemoryAddress	データ出力エリア アドレス	200
handshake	出力制御 0：なし 1：ハンドシェイク	1
responseTimeout	リトライ間隔[ms]	10000

・エリア種別一覧

エリア種別名	エリア種別番号
データレジスタ	168
ファイルレジスタ	175
リンクレジスタ	180

・サポートする機能

IoInitialize	OK	-
GetPort	-	-
PutPort	-	-
GetAll	-	-
PutAll	-	-

BusyOut	-	-
JudgeOut	-	-
RunOut	-	-
ReceiveData	-	-
SendData	OK	データを送信します。（参照：「SendData (P.8-246)」）
SendString	-	-
ReadPlcMemory	OK	PLC のメモリ領域から値を読み込みます。（参照：「ReadPlcMemory (P.8-210)」）
WritePlcMemory	OK	PLC のメモリ領域に値を書き込みます。（参照：「WritePlcMemory (P.8-329)」）
SetPlcData	OK	WritePlcMemory 関数で書き込むデータを作成します。（参照：「SetPlcData (P.8-258)」）
GetPlcData	OK	ReadPlcMemory 関数で読み込んだデータを取り出します。（参照：「GetPlcData (P.8-126)」）
SetPollingState	OK	通信モジュールの実行状態を設定します。（参照：「SetPollingState (P.8-260)」）
GetPollingState	OK	通信モジュールの実行状態を取得します。（参照：「GetPollingState (P.8-127)」）
ErrorOut	-	-

・使用例

SendData を使用してデータを送信します。

```
Dim data&(256)

' 整数型データ 5 つ分のデータを送信します
SendData "SerialPlcLinkM", data&(), 4*5
```

9-4-7 SerialPlcLink

PLC リンク(SYSMAC CS/CJ/CP/One)

・IoModule 識別名
SerialPlcLink

・概要
シリアル PLC リンクインタフェースを用いて、コマンドおよびデータの送受信を行います。

・システムデータ設定項目

識別名	意味	初期値
rsMode	通信種別 0：RS-232C	0
baudRate	通信速度[bps]	9600
byteSize	データ長[bit] (7 or 8)	7
parity	パリティ 0：なし 1：奇数 2：偶数	2
stopBits	ストップビット[bit] 0：1 1：2	1
softFlow	フロー制御 0：なし 1：あり（Xon/Xoff）	0
timeout	タイムアウト[s]	5

・PLC リンク設定項目

識別名	意味	初期値
commandArea	命令エリア エリア種別	チャネル IO(CIO)
commandMemoryAddress	命令エリア アドレス	0
responseArea	応答エリア エリア種別	チャネル IO(CIO)
responseMemoryAddress	応答エリア アドレス	100
outputArea	データ出力エリア エリア種別	チャネル IO(CIO)
outputMemoryAddress	データ出力エリア アドレス	200
handshake	出力制御 0：なし 1：ハンドシェイク	1
responseTimeout	リトライ間隔[ms]	10000

・エリア種別一覧

エリア種別名	エリア種別番号
チャネル IO(CIO)	176
内部補助リレー(WR)	177
保持リレー(HR)	178
特殊保持リレー(AR)	179
データメモリ(DM)	130
拡張データメモリ(EM0)	160
拡張データメモリ(EM1)	161
拡張データメモリ(EM2)	162
拡張データメモリ(EM3)	163
拡張データメモリ(EM4)	164

エリア種別名	エリア種別番号
拡張データメモリ(EM5)	165
拡張データメモリ(EM6)	166
拡張データメモリ(EM7)	167
拡張データメモリ(EM8)	168
拡張データメモリ(EM9)	169
拡張データメモリ(EMA)	170
拡張データメモリ(EMB)	171
拡張データメモリ(EMC)	172

・サポートする機能

IoInitialize	OK	-
GetPort	-	-
PutPort	-	-
GetAll	-	-
PutAll	-	-
BusyOut	-	-
JudgeOut	-	-
RunOut	-	-
ReceiveData	-	-
SendData	OK	データを送信します。(参照：「SendData (P.8-246)」)
SendString	-	-
ReadPlcMemory	OK	PLC のメモリ領域から値を読み込みます。(参照：「ReadPlcMemory (P.8-210)」)
WritePlcMemory	OK	PLC のメモリ領域に値を書き込みます。(参照：「WritePlcMemory (P.8-329)」)
SetPlcData	OK	WritePlcMemory 関数で書き込むデータを作成します。(参照：「SetPlcData (P.8-258)」)
GetPlcData	OK	ReadPlcMemory 関数で読み込んだデータを取り出します。(参照：「GetPlcData (P.8-126)」)
SetPollingState	OK	通信モジュールの実行状態を設定します。(参照：「SetPollingState (P.8-260)」)
GetPollingState	OK	通信モジュールの実行状態を取得します。(参照：「GetPollingState (P.8-127)」)
ErrorOut	-	-

・使用例

SendData を使用してデータを送信します。

```
Dim data&(256)

' 整数型データ 5 つ分のデータを送信します
SendData "SerialPlcLink", data&(), 4*5
```

ReadPlcMemory を使用して DM エリアの 10ch から 7ch データを取得します。
GetPlcData コマンドを使用して readData() からデータを抽出します。

```
Dim readData&(256)
Dim data3$(21)

'PLC よりデータを取得します
ReadPlcMemory "SerialPlcLink", 130, 10, 7, readData&()

' 取得データより、実数型のデータを抽出します
GetPlcData "SerialPlcLink", readData&(), 0, 8, data1#
' 取得データより、整数型のデータを抽出します
GetPlcData "SerialPlcLink", readData&(), 8, 4, data2&
' 取得データより、文字列型のデータを抽出します
GetPlcData "SerialPlcLink", readData&(), 12, 5, data2&
```

WritePlcMemory を使用して DM エリアの 10ch から 7ch 分のデータを書き込みます。
writeData() には、SetPlcData コマンドを使用してデータを設定します。

```
Dim writeData&(256)

'writeData に実数型のデータ(123.45) を設定します
SetPlcData "SerialPlcLink", writeData&(), 0, 8, 123.45
'writeData に整数型のデータ(20) を設定します
SetPlcData "SerialPlcLink", writeData&(), 32, 4, 20
'writeData に文字列型のデータ(OMRON) を設定します
SetPlcData "SerialPlcLink", writeData&(), 36, 5, "OMRON"

'DM エリアの 10ch から 7ch データを書き込みます
WritePlcMemory "SerialPlcLink", 130, 10, 7, writeData&()
```

9-4-8 TcpClient

無手順（TCP クライアント）

・IoModule 識別名

TcpClient

・概要

Ethernet の TCP クライアントプロトコルを用いて、コマンドおよびデータの送受信を行います。

・システムデータ設定項目

識別名	意味	初期値
enableDhcp	アドレス設定 0：次の IP アドレスを使う 1：IP アドレスを自動的に取得する	0
ipAddress	IP アドレス	10.5.5.100
subnetMask	サブネットマスク	255.255.255.0
defaultGateway	デフォルトゲートウェイ	10.5.5.110
dns	DNS サーバ	10.5.5.1
serverIpAddress	接続先 IP アドレス	10.5.5.101
portNo	ポート番号	9600

・サポートする機能

IoInitialize	OK	-
GetPort	-	-
PutPort	-	-
GetAll	-	-
PutAll	-	-
BusyOut	-	-
JudgeOut	-	-
RunOut	-	-
ReceiveData	OK	データを受信します。（参照：「ReceiveData (P.8-212)」）
SendData	OK	データを送信します。（参照：「SendData (P.8-246)」）
SendString	OK	文字列データを送信します。（参照：「SendString (P.8-248)」）
ReadPlcMemory	-	-
WritePlcMemory	-	-
SetPlcData	-	-
GetPlcData	-	-
SetPollingState	OK	通信モジュールの実行状態を設定します。（参照：「SetPollingState (P.8-260)」）
GetPollingState	OK	通信モジュールの実行状態を取得します。（参照：「GetPollingState (P.8-127)」）
ErrorOut	-	-

・使用例

ReceiveData を使用してデータを受信します。
Ethernet を使用するため、パラメータに IP アドレスとパラメータサイズ（整数型 × 4 領域）を設定します。

```
Dim data&(256)
Dim ipaddr&(4)

' 出力先の IP アドレスを設定します
ipaddr&(0) = 10
ipaddr&(1) = 1
ipaddr&(2) = 1
ipaddr&(3) = 101

' 整数型データ 5 つ分のデータを取得します
ReceiveData "TcpClient", data&(), 4*5, size&, ipaddr&(), 4*4
```

SendData を使用してデータを送信します。
Ethernet を使用するため、パラメータに IP アドレスとパラメータサイズ（整数型 × 4 領域）を設定します。

```
Dim data&(256)
Dim ipaddr&(4)

' 出力先の IP アドレスを設定します
ipaddr&(0) = 10
ipaddr&(1) = 1
ipaddr&(2) = 1
ipaddr&(3) = 101

' 整数型データ 5 つ分のデータを送信します
SendData "TcpClient", data&(), 4*5, ipaddr&(), 4*4
```

9-4-9 TcpNormal

無手順(TCP) 通信

・IoModule 識別名

TcpNormal

・概要

Ethernet の TCP サーバを用いて、コマンドおよびデータの送受信を行います。

・システムデータ設定項目

識別名	意味	初期値
enableDhcp	アドレス設定 0：次の IP アドレスを使う 1：IP アドレスを自動的に取得する	0
ipAddress	IP アドレス	10.5.5.100
subnetMask	サブネットマスク	255.255.255.0
defaultGateway	デフォルトゲートウェイ	10.5.5.110
dns	DNS サーバ	10.5.5.1
portNo	ポート番号	9600

・サポートする機能

IoInitialize	OK	-
GetPort	-	-
PutPort	-	-
GetAll	-	-
PutAll	-	-
BusyOut	-	-
JudgeOut	-	-
RunOut	-	-
ReceiveData	OK	データを受信します。（参照：「ReceiveData (P.8-212)」）
SendData	OK	データを送信します。（参照：「SendData (P.8-246)」）
SendString	OK	文字列データを送信します。（参照：「SendString (P.8-248)」）
ReadPlcMemory	-	-
WritePlcMemory	-	-
SetPlcData	-	-
GetPlcData	-	-
SetPollingState	OK	通信モジュールの実行状態を設定します。（参照：「SetPollingState (P.8-260)」）
GetPollingState	OK	通信モジュールの実行状態を取得します。（参照：「GetPollingState (P.8-127)」）
ErrorOut	-	-

・使用例

ReceiveData を使用してデータを受信します。
Ethernet を使用するため、パラメータに IP アドレスとパラメータサイズ（整数型 × 4 領域）を設定します。

```
Dim data&(256)
Dim ipaddr&(4)

' 出力先の IP アドレスを設定します
ipaddr&(0) = 10
ipaddr&(1) = 1
ipaddr&(2) = 1
ipaddr&(3) = 101

' 整数型データ 5 つ分のデータを取得します
ReceiveData "TcpNormal", data&(), 4*5, size&, ipaddr&(), 4*4
```

SendData を使用してデータを送信します。
Ethernet を使用するため、パラメータに IP アドレスとパラメータサイズ（整数型 × 4 領域）を設定します。

```
Dim data&(256)
Dim ipaddr&(4)

' 出力先の IP アドレスを設定します
ipaddr&(0) = 10
ipaddr&(1) = 1
ipaddr&(2) = 1
ipaddr&(3) = 101

' 整数型データ 5 つ分のデータを送信します
SendData "TcpNormal", data&(), 4*5, ipaddr&(), 4*4
```

9-4-10 UdpNormal

UDP 無手順通信

・IoModule 識別名

UdpNormal
UdpNormal2（Fxxx シリーズ方式）

・概要

Ethernet の UDP プロトコルを用いて、コマンドおよびデータの送受信を行います。

・システムデータ設定項目

識別名	意味	初期値
enableDhcp	アドレス設定 0：次の IP アドレスを使う 1：IP アドレスを自動的に取得する	0
ipAddress	IP アドレス	10.5.5.100
subnetMask	サブネットマスク	255.255.255.0
defaultGateway	デフォルトゲートウェイ	10.5.5.110
dns	DNS サーバ	10.5.5.1
destIpAddress	出力 IP アドレス	0.0.0.0
portNo	入力ポート番号	9600
portNo2	出力ポート番号*1	-1

*1. 入力ポート番号と出力ポート番号が同じ場合、出力ポート番号に-1 を設定します。

・サポートする機能

IoInitialize	OK	-
GetPort	-	-
PutPort	-	-
GetAll	-	-
PutAll	-	-
BusyOut	-	-
JudgeOut	-	-
RunOut	-	-
ReceiveData	OK	データを受信します。（参照：「ReceiveData (P.8-212)」）
SendData	OK	データを送信します。（参照：「SendData (P.8-246)」）
SendString	OK	文字列データを送信します。（参照：「SendString (P.8-248)」）
ReadPlcMemory	-	-
WritePlcMemory	-	-
SetPlcData	-	-
GetPlcData	-	-
SetPollingState	OK	通信モジュールの実行状態を設定します。（参照：「SetPollingState (P.8-260)」）
GetPollingState	OK	通信モジュールの実行状態を取得します。（参照：「GetPollingState (P.8-127)」）
ErrorOut	-	-

・使用例

ReceiveData を使用してデータを受信します。
Ethernet を使用するため、パラメータに IP アドレスとパラメータサイズ（整数型 × 4 領域）を設定します。

```
Dim data&(256)
Dim ipaddr&(4)

' 出力先の IP アドレスを設定します
ipaddr&(0) = 10
ipaddr&(1) = 1
ipaddr&(2) = 1
ipaddr&(3) = 101

' 整数型データ 5 つ分のデータを取得します
ReceiveData "UdpNormal", data&(), 4*5, size&, ipaddr&(), 4*4
```

SendData を使用してデータを送信します。
Ethernet を使用するため、パラメータに IP アドレスとパラメータサイズ（整数型 × 4 領域）を設定します。

```
Dim data&(256)
Dim ipaddr&(4)

' 出力先の IP アドレスを設定します
ipaddr&(0) = 10
ipaddr&(1) = 1
ipaddr&(2) = 1
ipaddr&(3) = 101

' 整数型データ 5 つ分のデータを送信します
SendData "UdpNormal", data&(), 4*5, ipaddr&(), 4*4
```

9-4-11 UdpPlcLinkM

PLC リンク(MELSEC QnU/Q/QnAS)(UDP)

・IoModule 識別名

UdpPlcLinkM

・概要

EtherNetPLC リンクインタフェースを用いて、コマンドおよびデータの送受信を行います。

・システムデータ設定項目

識別名	意味	初期値
enableDhcp	アドレス設定 0：次の IP アドレスを使う 1：IP アドレスを自動的に取得する	0
ipAddress	IP アドレス	10.5.5.100
subnetMask	サブネットマスク	255.255.255.0
defaultGateway	デフォルトゲートウェイ	10.5.5.110
dns	DNS サーバ	10.5.5.1
destIpAddress	出力 IP アドレス	0.0.0.0
portNo	ポート番号	9600

・PLC リンク設定項目

識別名	意味	初期値
commandArea	命令エリア エリア種別	データレジスタ
commandMemoryAddress	命令エリア アドレス	0
responseArea	応答エリア エリア種別	データレジスタ
responseMemoryAddress	応答エリア アドレス	100
outputArea	データ出力エリア エリア種別	データレジスタ
outputMemoryAddress	データ出力エリア アドレス	200
handshake	出力制御 0：なし 1：ハンドシェイク	1
responseTimeout	リトライ間隔[ms]	10000
responseTimeout2	リトライ間隔 2[ms]	1000

・エリア種別一覧

エリア種別名	エリア種別番号
データレジスタ	168
ファイルレジスタ	175
リンクレジスタ	180

・サポートする機能

IoInitialize	OK	-
GetPort	-	-
PutPort	-	-
GetAll	-	-
PutAll	-	-
BusyOut	-	-
JudgeOut	-	-
RunOut	-	-
ReceiveData	-	-

SendData	OK	データを送信します。（参照：「SendData (P.8-246)」）
SendString	-	-
ReadPlcMemory	OK	PLC のメモリ領域から値を読み込みます。（参照：「ReadPlcMemory (P.8-210)」）
WritePlcMemory	OK	PLC のメモリ領域に値を書き込みます。（参照：「WritePlcMemory (P.8-329)」）
SetPlcData	OK	WritePlcMemory 関数で書き込むデータを作成します。（参照：「SetPlcData (P.8-258)」）
GetPlcData	OK	ReadPlcMemory 関数で読み込んだデータを取り出します。（参照：「GetPlcData (P.8-126)」）
SetPollingState	OK	通信モジュールの実行状態を設定します。（参照：「SetPollingState (P.8-260)」）
GetPollingState	OK	通信モジュールの実行状態を取得します。（参照：「GetPollingState (P.8-127)」）
ErrorOut	-	-

・使用例

SendData を使用してデータを送信します。
Ethernet を使用するため、パラメータに IP アドレスとパラメータサイズ（整数型 × 4 領域）を設定します。

```
Dim data&(256)
Dim ipaddr&(4)

' 出力先の IP アドレスを設定します
ipaddr&(0) = 10
ipaddr&(1) = 1
ipaddr&(2) = 1
ipaddr&(3) = 101

' 整数型データ 5 つ分のデータを送信します
SendData "UdpPlcLinkM", data&(), 4*5, ipaddr&(), 4*4
```

ReadPlcMemory を使用してデータレジスタエリアの 10ch から 7ch データを取得します。
GetPlcData コマンドを使用して readData() からデータを抽出します。

```
Dim readData&(256)
Dim data3$(21)
'PLC のデータレジスタエリアよりデータを取得します
ReadPlcMemory "UdpPlcLinkM", 176, 10, 7, readData&()

' 取得データより、実数型のデータを抽出します
GetPlcData "UdpPlcLinkM", readData&(), 0, 8, data1#
' 取得データより、整数型のデータを抽出します
GetPlcData "UdpPlcLinkM", readData&(), 8, 4, data2&
' 取得データより、文字列型のデータを抽出します
GetPlcData "UdpPlcLinkM", readData&(), 12, 5, data2&
```

WritePlcMemory を使用してデータレジスタエリアの 10ch から 7ch 分のデータを書き込みます。
writeData() には、SetPlcData コマンドを使用してデータを設定します。

```
Dim writeData&(256)

'writeData に実数型のデータ(123.45) を設定します
SetPlcData "UdpPlcLinkM", writeData&(), 0, 8, 123.45
'writeData に整数型のデータ(20) を設定します
SetPlcData "UdpPlcLinkM", writeData&(), 32, 4, 20
'writeData に文字列型のデータ(OMRON) を設定します
SetPlcData "UdpPlcLinkM", writeData&(), 36, 5, "OMRON"

' データレジスタエリアの 10ch から 7ch データを書き込みます
WritePlcMemory "UdpPlcLinkM", 168, 10, 7, writeData&()
```

9-4-12 UdpPlcLinkY

PLC リンク(JEPMC MP)

・IoModule 識別名
UdpPlcLinkY

・概要
EtherNetPLC リンクインタフェースを用いて、コマンドおよびデータの送受信を行います。

・システムデータ設定項目

識別名	意味	初期値
enableDhcp	アドレス設定 0：次の IP アドレスを使う 1：IP アドレスを自動的に取得する	0
ipAddress	IP アドレス	10.5.5.100
subnetMask	サブネットマスク	255.255.255.0
defaultGateway	デフォルトゲートウェイ	10.5.5.110
dns	DNS サーバ	10.5.5.1
destIpAddress	出力 IP アドレス	0.0.0.0
portNo	ポート番号	9600

・PLC リンク設定項目

識別名	意味	初期値
commandArea	命令エリア エリア種別	データレジスタ
commandMemoryAddress	命令エリア アドレス	0
responseArea	応答エリア エリア種別	データレジスタ
responseMemoryAddress	応答エリア アドレス	100
outputArea	データ出力エリア エリア種別	データレジスタ
outputMemoryAddress	データ出力エリア アドレス	200
handshake	出力制御 0：なし 1：ハンドシェイク	1
responseTimeout	リトライ間隔[ms]	10000
responseTimeout2	リトライ間隔 2[ms]	1000

・エリア種別一覧

エリア種別名	エリア種別番号
データレジスタ	176

・サポートする機能

IoInitialize	OK	-
GetPort	-	-
PutPort	-	-
GetAll	-	-
PutAll	-	-
BusyOut	-	-
JudgeOut	-	-
RunOut	-	-
ReceiveData	-	-
SendData	OK	データを送信します。(参照：「SendData (P.8-246)」)
SendString	-	-

ReadPlcMemory	OK	PLC のメモリ領域から値を読み込みます。（参照：「ReadPlcMemory (P.8-210)」）
WritePlcMemory	OK	PLC のメモリ領域に値を書き込みます。（参照：「WritePlcMemory (P.8-329)」）
SetPlcData	OK	WritePlcMemory 関数で書き込むデータを作成します。（参照：「SetPlcData (P.8-258)」）
GetPlcData	OK	ReadPlcMemory 関数で読み込んだデータを取り出します。（参照：「GetPlcData (P.8-126)」）
SetPollingState	OK	通信モジュールの実行状態を設定します。（参照：「SetPollingState (P.8-260)」）
GetPollingState	OK	通信モジュールの実行状態を取得します。（参照：「GetPollingState (P.8-127)」）
ErrorOut	-	-

・使用例

SendData を使用してデータを送信します。
Ethernet を使用するため、パラメータに IP アドレスとパラメータサイズ（整数型 × 4 領域）を設定します。

```
Dim data&(256)
Dim ipaddr&(4)

' 出力先の IP アドレスを設定します
ipaddr&(0) = 10
ipaddr&(1) = 1
ipaddr&(2) = 1
ipaddr&(3) = 101

' 整数型データ 5 つ分のデータを送信します
SendData "UdpPlcLinkY", data&(), 4*5, ipaddr&(), 4*4
```

ReadPlcMemory を使用してデータレジスタエリアの 10ch から 7ch データを取得します。
GetPlcData コマンドを使用して readData() からデータを抽出します。

```
Dim readData&(256)
Dim data3$(21)

'PLC のデータレジスタエリアよりデータを取得します
ReadPlcMemory "UdpPlcLinkY", 168, 10, 7, readData&()

' 取得データより、実数型のデータを抽出します
GetPlcData "UdpPlcLinkY", readData&(), 0, 8, data1#
' 取得データより、整数型のデータを抽出します
GetPlcData "UdpPlcLinkY", readData&(), 8, 4, data2&
' 取得データより、文字列型のデータを抽出します
GetPlcData "UdpPlcLinkY", readData&(), 12, 5, data2&
```

WritePlcMemory を使用してデータレジスタエリアの 10ch から 7ch 分のデータを書き込みます。
writeData() には、SetPlcData コマンドを使用してデータを設定します。

```
Dim writeData&(256)

'writeData に実数型のデータ(123.45) を設定します
SetPlcData "UdpPlcLinkY", writeData&(), 0, 8, 123.45
'writeData に整数型のデータ(20) を設定します
SetPlcData "UdpPlcLinkY", writeData&(), 32, 4, 20
'writeData に文字列型のデータ(OMRON) を設定します
SetPlcData "UdpPlcLinkY", writeData&(), 36, 5, "OMRON"

' データレジスタエリアの 10ch から 7ch データを書き込みます
WritePlcMemory "UdpPlcLinkY", 176, 10, 7, writeData&()
```

9-4-13 UdpPlcLink

PLC リンク(SYSMAC CS/CJ/CP/One)(UDP)

・IoModule 識別名

UdpPlcLink

・概要

EtherNetPLC リンクインタフェースを用いて、コマンドおよびデータの送受信を行います。

・システムデータ設定項目

識別名	意味	初期値
enableDhcp	アドレス設定 0：次の IP アドレスを使う 1：IP アドレスを自動的に取得する	0
ipAddress	IP アドレス	10.5.5.100
subnetMask	サブネットマスク	255.255.255.0
defaultGateway	デフォルトゲートウェイ	10.5.5.110
dns	DNS サーバ	10.5.5.1
destIpAddress	出力 IP アドレス	0.0.0.0
portNo	ポート番号	9600

・PLC リンク設定項目

識別名	意味	初期値
commandArea	命令エリア エリア種別	チャネル IO(CIO)
commandMemoryAddress	命令エリア アドレス	0
responseArea	応答エリア エリア種別	チャネル IO(CIO)
responseMemoryAddress	応答エリア アドレス	100
outputArea	データ出力エリア エリア種別	チャネル IO(CIO)
outputMemoryAddress	データ出力エリア アドレス	200
handshake	出力制御 0：なし 1：ハンドシェイク	1
responseTimeout	リトライ間隔[ms]	10000
responseTimeout2	リトライ間隔 2[ms]	1000

・エリア種別一覧

エリア種別名	エリア種別番号
チャネル IO(CIO)	176
内部補助リレー(WR)	177
保持リレー(HR)	178
特殊保持リレー(AR)	179
データメモリ(DM)	130
拡張データメモリ(EM0)	160
拡張データメモリ(EM1)	161
拡張データメモリ(EM2)	162
拡張データメモリ(EM3)	163
拡張データメモリ(EM4)	164
拡張データメモリ(EM5)	165
拡張データメモリ(EM6)	166
拡張データメモリ(EM7)	167
拡張データメモリ(EM8)	168

エリア種別名	エリア種別番号
拡張データメモリ(EM9)	169
拡張データメモリ(EMA)	170
拡張データメモリ(EMB)	171
拡張データメモリ(EMC)	172

・サポートする機能

IoInitialize	OK	-
GetPort	-	-
PutPort	-	-
GetAll	-	-
PutAll	-	-
BusyOut	-	-
JudgeOut	-	-
RunOut	-	-
ReceiveData	-	-
SendData	OK	データを送信します。(参照：「SendData (P.8-246)」)
SendString	-	-
ReadPlcMemory	OK	PLC のメモリ領域から値を読み込みます。(参照：「ReadPlcMemory (P.8-210)」)
WritePlcMemory	OK	PLC のメモリ領域に値を書き込みます。(参照：「WritePlcMemory (P.8-329)」)
SetPlcData	OK	WritePlcMemory 関数で書き込むデータを作成します。(参照：「SetPlcData (P.8-258)」)
GetPlcData	OK	ReadPlcMemory 関数で読み込んだデータを取り出します。(参照：「GetPlcData (P.8-126)」)
SetPollingState	OK	通信モジュールの実行状態を設定します。(参照：「SetPollingState (P.8-260)」)
GetPollingState	OK	通信モジュールの実行状態を取得します。(参照：「GetPollingState (P.8-127)」)
ErrorOut	-	-

・使用例

SendData を使用してデータを送信します。
Ethernet を使用するため、パラメータに IP アドレスとパラメータサイズ（整数型 × 4 領域）を設定します。

```
Dim data&(256)
Dim ipaddr&(4)

' 出力先の IP アドレスを設定します
ipaddr&(0) = 10
ipaddr&(1) = 1
ipaddr&(2) = 1
ipaddr&(3) = 101

' 整数型データ 5 つ分のデータを送信します
SendData "UdpPlcLink", data&(), 4*5, ipaddr&(), 4*4
```

ReadPlcMemory を使用して DM エリアの 10ch から 7ch データを取得します。
GetPlcData コマンドを使用して readData() からデータを抽出します。

```
Dim readData&(256)
Dim data3$(21)

'PLC よりデータを取得します
ReadPlcMemory "UdpPlcLink", 130, 10, 7, readData&()

' 取得データより、実数型のデータを抽出します
GetPlcData "UdpPlcLink", readData&(), 0, 8, data1#
' 取得データより、整数型のデータを抽出します
GetPlcData "UdpPlcLink", readData&(), 8, 4, data2&
'' 取得データより、文字列型のデータを抽出します
GetPlcData "UdpPlcLink", readData&(), 12, 5, data2&
```

WritePlcMemory を使用して DM エリアの 10ch から 7ch 分のデータを書き込みます。
writeData() には、SetPlcData コマンドを使用してデータを設定します。

```
Dim writeData&(256)

'writeData に実数型のデータ(123.45) を設定します
SetPlcData "UdpPlcLink", writeData&(), 0, 8, 123.45
'writeData に整数型のデータ(20) を設定します
SetPlcData "UdpPlcLink", writeData&(), 32, 4, 20
'writeData に文字列型のデータ(OMRON) を設定します
SetPlcData "UdpPlcLink", writeData&(), 36, 5, "OMRON"

'DM エリアの 10ch から 7ch データを書き込みます
WritePlcMemory "UdpPlcLink", 130, 10, 7, writeData&()
```

9-4-14 TcpPlcLinkM

PLC リンク(MELSEC QnU/Q/QnAS)(TCP)

・IoModule 識別名

TcpPlcLinkM

・概要

EtherNetPLC リンクインタフェースを用いて、コマンドおよびデータの送受信を行います。

・システムデータ設定項目

識別名	意味	初期値
enableDhcp	アドレス設定 0：次の IP アドレスを使う 1：IP アドレスを自動的に取得する	0
ipAddress	IP アドレス	10.5.5.100
subnetMask	サブネットマスク	255.255.255.0
defaultGateway	デフォルトゲートウェイ	10.5.5.110
dns	DNS サーバ	10.5.5.1
wins	WINS サーバ	0.0.0.0, 0.0.0.0*1
enableDhcp2	アドレス設定 0：次の IP アドレスを使う 1：IP アドレスを自動的に取得する	0
ipAddress2	IP アドレス 2	10.5.6.100
subnetMask2	サブネットマスク 2	255.255.255.0
defaultGateway2	デフォルトゲートウェイ 2	10.5.6.100
dns2	DNS サーバ 2	10.5.6.100
wins2	WINS サーバ 2	0.0.0.0, 0.0.0.0*1
destIpAddress	出力 IP アドレス	10.5.5.101
portNo	ポート番号	9876

*1. ［外部ツール］メニューから、［システム設定］の［通信］の設定画面表示上では、「0.0.0.0」ですが、空データになります。

・PLC リンク設定項目

識別名	意味	初期値
commandArea	命令エリア エリア種別	データレジスタ
commandMemoryAddress	命令エリア アドレス	0
responseArea	応答エリア エリア種別	データレジスタ
responseMemoryAddress	応答エリア アドレス	100
outputArea	データ出力エリア エリア種別	データレジスタ
outputMemoryAddress	データ出力エリア アドレス	200
handshake	出力制御 0：なし 1：ハンドシェイク	1
responseTimeout	リトライ間隔[ms]	10000
outputBuffering	非同期出力 0：OFF 1：ON	0
	ポーリング周期[ms]	0

・エリア種別一覧

エリア種別名	エリア種別番号
データレジスタ	168

エリア種別名	エリア種別番号
ファイルレジスタ	175
リンクレジスタ	180

・サポートする機能

IoInitialize	OK	-
GetPort	OK	指定した入力端子の入力状態を取得します。(参照：「GetPort (P.8-129)」)
PutPort	OK	出力端子に 1 点出力を行います。(参照：「PutPort (P.8-205)」)
GetAll	OK	入力端子の全点入力を行います。(参照：「GetAll (P.8-120)」)
PutAll	OK	出力端子に全点出力を行います。(参照：「PutAll (P.8-204)」)
BusyOut	-	-
JudgeOut	-	-
RunOut	-	-
ReceiveData	-	-
SendData	OK	データを送信します。(参照：「SendData (P.8-246)」)
SendString	-	-
ReadPlcMemory	OK	PLC のメモリ領域から値を読み込みます。(参照：「ReadPlcMemory (P.8-210)」)
WritePlcMemory	OK	PLC のメモリ領域に値を書き込みます。(参照：「WritePlcMemory (P.8-329)」)
SetPlcData	OK	WritePlcMemory 関数で書き込むデータを作成します。(参照：「SetPlcData (P.8-258)」)
GetPlcData	OK	ReadPlcMemory 関数で読み込んだデータを取り出します。(参照：「GetPlcData (P.8-126)」)
SetPollingState	OK	通信モジュールの実行状態を設定します。(参照：「SetPollingState (P.8-260)」)
GetPollingState	OK	通信モジュールの実行状態を取得します。(参照：「GetPollingState (P.8-127)」)
ErrorOut	-	-

・使用例

SendData を使用してデータを送信します。
Ethernet を使用するため、パラメータに IP アドレスとパラメータサイズ（整数型 × 4 領域）を設定します。

```
Dim data&(256)
Dim ipaddr&(4)

' 出力先の IP アドレスを設定します
ipaddr&(0) = 10
ipaddr&(1) = 1
ipaddr&(2) = 1
ipaddr&(3) = 101

' 整数型データ 5 つ分のデータを送信します
SendData "TcpPlcLinkM", data&(), 4*5, ipaddr&(), 4*4
```

ReadPlcMemory を使用してデータレジスタエリアの 10ch から 7ch データを取得します。
GetPlcData コマンドを使用して readData() からデータを抽出します。

```
Dim readData&(256)
Dim data3$

'PLC のデータレジスタエリアよりデータを取得します
ReadPlcMemory "TcpPlcLinkM", 168, 10, 7, readData&()

' 取得データより、実数型のデータを抽出します
GetPlcData "TcpPlcLinkM", readData&(), 0, 8, data1#
' 取得データより、整数型のデータを抽出します
GetPlcData "TcpPlcLinkM", readData&(), 8, 4, data2&
' 取得データより、文字列型のデータを抽出します
GetPlcData "TcpPlcLinkM", readData&(), 12, 5, data3$
```

WritePlcMemory を使用してデータレジスタエリアの 10ch から 7ch 分のデータを書き込みます。writeData() には、SetPlcData コマンドを使用してデータを設定します。

```
Dim writeData&(256)

'writeData に実数型のデータ(123.45) を設定します
SetPlcData "TcpPlcLinkM", writeData&(), 0, 8, 123.45
'writeData に整数型のデータ(20) を設定します
SetPlcData "TcpPlcLinkM", writeData&(), 32, 4, 20
'writeData に文字列型のデータ(OMRON) を設定します
SetPlcData "TcpPlcLinkM", writeData&(), 36, 5, "OMRON"

' データレジスタエリアの 10ch から 7ch データを書き込みます
WritePlcMemory "TcpPlcLinkM", 168, 10, 7, writeData&()
```

9-4-15 TcpPlcLink

PLC リンク(SYSMAC CS/CJ/CP/One)(TCP)

・IoModule 識別名

TcpPlcLink

・概要

EtherNetPLC リンクインタフェースを用いて、コマンドおよびデータの送受信を行います。

・システムデータ設定項目

識別名	意味	初期値
enableDhcp	アドレス設定 0：次の IP アドレスを使う 1：IP アドレスを自動的に取得する	0
ipAddress	IP アドレス	10.5.5.100
subnetMask	サブネットマスク	255.255.255.0
defaultGateway	デフォルトゲートウェイ	10.5.5.110
dns	DNS サーバ	10.5.5.1
wins	WINS サーバ	0.0.0.0, 0.0.0.0*1
enableDhcp2	アドレス設定 0：次の IP アドレスを使う 1：IP アドレスを自動的に取得する	0
ipAddress2	IP アドレス 2	10.5.6.100
subnetMask2	サブネットマスク 2	255.255.255.0
defaultGateway2	デフォルトゲートウェイ 2	10.5.6.100
dns2	DNS サーバ 2	10.5.6.100
wins2	WINS サーバ 2	0.0.0.0, 0.0.0.0*1
destIpAddress	出力 IP アドレス	10.5.5.101
portNo	ポート番号	9876

*1. ［外部ツール］メニューから、［システム設定］の［通信］の設定画面表示上では、「0.0.0.0」ですが、空データになります。

・PLC リンク設定項目

識別名	意味	初期値
commandArea	命令エリア エリア種別	チャネル IO(CIO)
commandMemoryAddress	命令エリア アドレス	0
responseArea	応答エリア エリア種別	チャネル IO(CIO)
responseMemoryAddress	応答エリア アドレス	100
outputArea	データ出力エリア エリア種別	チャネル IO(CIO)
outputMemoryAddress	データ出力エリア アドレス	200
handshake	出力制御 0：なし 1：ハンドシェイク	1
responseTimeout	リトライ間隔[ms]	10000
outputBuffering	非同期出力 0：OFF 1：ON	0
pollingMinCycle	ポーリング周期[ms]	0

・エリア種別一覧

エリア種別名	エリア種別番号
チャネル IO(CIO)	176

エリア種別名	エリア種別番号
内部補助リレー(WR)	177
保持リレー(HR)	178
特殊保持リレー(AR)	179
データメモリ(DM)	130
拡張データメモリ(EM0)	160
拡張データメモリ(EM1)	161
拡張データメモリ(EM2)	162
拡張データメモリ(EM3)	163
拡張データメモリ(EM4)	164
拡張データメモリ(EM5)	165
拡張データメモリ(EM6)	166
拡張データメモリ(EM7)	167
拡張データメモリ(EM8)	168
拡張データメモリ(EM9)	169
拡張データメモリ(EMA)	170
拡張データメモリ(EMB)	171
拡張データメモリ(EMC)	172

・サポートする機能

IoInitialize	OK	-
GetPort	OK	指定した入力端子の入力状態を取得します。(参照：「GetPort (P.8-129)」)
PutPort	OK	出力端子に 1 点出力を行います。(参照：「PutPort (P.8-205)」)
GetAll	OK	入力端子の全点入力を行います。(参照：「GetAll (P.8-120)」)
PutAll	OK	出力端子に全点出力を行います。(参照：「PutAll (P.8-204)」)
BusyOut	-	-
JudgeOut	-	-
RunOut	-	-
ReceiveData	-	-
SendData	OK	データを送信します。(参照：「SendData (P.8-246)」)
SendString	-	-
ReadPlcMemory	OK	PLC のメモリ領域から値を読み込みます。(参照：「ReadPlcMemory (P.8-210)」)
WritePlcMemory	OK	PLC のメモリ領域に値を書き込みます。(参照：「WritePlcMemory (P.8-329)」)
SetPlcData	OK	WritePlcMemory 関数で書き込むデータを作成します。(参照：「SetPlcData (P.8-258)」)
GetPlcData	OK	ReadPlcMemory 関数で読み込んだデータを取り出します。(参照：「GetPlcData (P.8-126)」)
SetPollingState	OK	通信モジュールの実行状態を設定します。(参照：「SetPollingState (P.8-260)」)
GetPollingState	OK	通信モジュールの実行状態を取得します。(参照：「GetPollingState (P.8-127)」)
ErrorOut	-	-

・使用例

SendData を使用してデータを送信します。
Ethernet を使用するため、パラメータに IP アドレスとパラメータサイズ（整数型 × 4 領域）を設定します。

```
Dim data&(256)
Dim ipaddr&(4)

' 出力先の IP アドレスを設定します
ipaddr&(0) = 10
ipaddr&(1) = 1
ipaddr&(2) = 1
ipaddr&(3) = 101

' 整数型データ 5 つ分のデータを送信します
SendData "TcpPlcLink", data&(), 4*5, ipaddr&(), 4*4
```

ReadPlcMemory を使用して DM エリアの 10ch から 7ch データを取得します。
GetPlcData コマンドを使用して readData() からデータを抽出します。

```
Dim readData&(256)
Dim data3$

'PLC よりデータを取得します
ReadPlcMemory "TcpPlcLink", 130, 10, 7, readData&()
' 取得データより、実数型のデータを抽出します
GetPlcData "TcpPlcLink", readData&(), 0, 8, data1#
' 取得データより、整数型のデータを抽出します
GetPlcData "TcpPlcLink", readData&(), 8, 4, data2&
' 取得データより、文字列型のデータを抽出します
GetPlcData "TcpPlcLink", readData&(), 12, 5, data3$
```

WritePlcMemory を使用して DM エリアの 10ch から 7ch 分のデータを書き込みます。
writeData() には、SetPlcData コマンドを使用してデータを設定します。

```
Dim writeData&(256)

'writeData に実数型のデータ(123.45) を設定します
SetPlcData "TcpPlcLink", writeData&(), 0, 8, 123.45
'writeData に整数型のデータ(20) を設定します
SetPlcData "TcpPlcLink", writeData&(), 32, 4, 20
'writeData に文字列型のデータ(OMRON) を設定します
SetPlcData "TcpPlcLink", writeData&(), 36, 5, "OMRON"

'DM エリアの 10ch から 7ch データを書き込みます
WritePlcMemory "TcpPlcLink", 130, 10, 7, writeData&()
```

9-5 処理項目識別名一覧

9-5-1 検査・計測をする

処理項目	識別名	処理項目	識別名
サーチ	Search	フレキシブルサーチ	FlexibleSearch
センシティブサーチ	SensitiveSearch	ECM サーチ	EcmSearch
形状サーチⅡ	ShapeSearch2	形状サーチⅢ	ShapeSearch3
EC コーナー	EcCorner	EC クロス	EcCross
仕分け	Classification	エッジ位置	EdgePosition
エッジ本数	EdgePitch	スキャンエッジ位置	ScanEdgePosition
スキャンエッジ幅	ScanEdgeWidth	円形スキャンエッジ位置	CircularScanEdgePosition
円形スキャンエッジ幅	CircularScanEdgeWidth	交点座標	Intersection
色平均・偏差	ColorData	面積重心	ColorArea
ラベリング	ColorLabeling	ラベルデータ	LabelData
キズ汚れ	Defect	高精度キズ汚れ	PreciseDefect
ファインマッチング	FineMatching	汎用文字検査	CharacterInspect
日付照合	DateVerification	モデル辞書	ModelDictionary
2 次元コードⅡ	2dcr2	2 次元コード	2dcrV400
バーコード	BarcodeCC	OCR ユーザ辞書	CharacterDictionary
OCR	CharacterRecognition	円形角度取得	CircleAngle
塗布剤途切れ検査	EmbrocationTrack	サーチⅡ	Search2
AI ファインマッチング	FineMatchingAnalysisAI		

9-5-2 画像を取り込む

処理項目	識別名	処理項目	識別名
カメラ画像入力	CameraImage	カメラ画像入力 FH	CameraImageFH
カメラ画像入力 FHV	CameraImageFHV	カメラ画像入力 HDR	CameraImageHDR
カメラ画像入力 HDR Lite	CameraImageHDRLite	照度差ステレオ画像入力	CameraImagePS
カメラ切替	CameraSwitch	計測画像切替	MeasureImageSwitch
多段撮像	MultiImaging	*多段撮像終了	MultiImagingEnd
多段撮像タスク	MultiImagingTask		

注：項目名の先頭に "*" がついている項目は、表示されません。

9-5-3 画像を補正する

処理項目	識別名	処理項目	識別名
位置ずれ修正	Scroll	計測前処理	Filtering
背景カット	FilterHighContrast	明度補正フィルタ	FilterShading
カラーグレーフィルタ	FilterColorGray	色抽出フィルタ	FilterExtractColor
色ムラカット	FilterEven	縞模様カットⅡ	FilterRank
円展開	TransPolar	台形歪み補正	TrapezoidalCorrection2
外部機器シミュレータ	MachineSimulator	差分抽出	DifferenceExtract
高機能前処理	FilterAdvanced	パノラマ	Panorama2
トリミング&リサイズ	TrimAndResize	画像連結	Stitch

9-5-4 検査・計測を補助する

処理項目	識別名	処理項目	識別名
ユニットマクロ	Macro	ユニット演算マクロ	MacroCalc
演算	Calculation	近似直線	ApproximateLine
近似円	ApproximateCircle	高精度キャリブレーション	CalibrationP
ユーザデータ	UserData	処理ユニットデータ設定	SetUnitData
処理ユニットデータ取得	GetUnitData	処理ユニット図形設定	SetUnitFigure
処理ユニット図形取得	GetUnitFigure	トレンドモニタ	TrendMonitor
画像ロギング	ImageLogging	画像変換ロギング	MeasureImageSave
データロギング	DataLogging	経過時間	ElapsedTime
ウェイト	Wait	フォーカス	SettingFocus
アイリス	SettingBright	並列化	Parallelize
*並列化終了	ParallelizeEnd	並列化タスク	ParallelizeTask
統計処理	Statistics	キャリブレーション参照	ReferenceCalibData
位置角度演算	SetPositionData	ステージデータ	StageData
ロボットデータ	RobotData	画像マスタキャリブレーション	AlignmentCalibration
PLC マスタキャリブレーション	AlignmentCalibrationPLC	位置角度変換	ScrollPositionData
軸移動量演算	ScrollCalculation	多点軸移動量演算	ScrollCalculationMulti
検出点	DetectionPoint	強制座標設定	ForcePositionSetting
カメラキャリブレーション	CameraCalibration	データ保存	DataSave
コンベアキャリブレーション	ConveyorCalibration	シーン	Scene
システム情報	SystemInfo		

注：項目名の先頭に "*" がついている項目は、表示されません。

9-5-5 処理を分岐する

処理項目	識別名	処理項目	識別名
条件分岐	Branch	計測終了	End
入力条件分岐	InputBranch	無手順フロー制御	ControlFlowStringCommand
PLC リンクフロー制御	ControlFlowPlcLinkCommand	パラレルフロー制御	ControlFlowParallelCommand
Fieldbus フロー制御	ControlFlowFieldbusCommand	条件実行 (If)	If
条件実行 (Else)	IfElse	*条件実行終了	IfEnd
ループ	Loop	ループ中断	LoopExit
*ループ終了	LoopEnd	選択実行 (Select)	Select
選択実行 (Case)	SelectCase	*選択実行終了	SelectEnd

注：項目名の先頭に "*" がついている項目は、表示されません。

9-5-6 結果を外部へ出力する

処理項目	識別名	処理項目	識別名
結果出力 (I/O)	OutputIO	結果出力（メッセージ）	OutputString
シリアルデータ出力	NormalData	パラレルデータ出力	ParallelData
パラレル判定出力	ParallelJudge	Fieldbus データ出力	FieldbusData
結果出力 (パラレル I/O)	OutputParallelIo		

9-5-7 結果を画面へ表示する

処理項目	識別名	処理項目	識別名
結果表示	ResultDisplay	画像ファイル表示	DisplayImageFile
最新 NG 画像表示	DisplayLastNG	コンベアパノラマ表示	ConveyorPanorama
表示画像保持	DisplayImage		

9-6 図形データ一覧

処理ユニットが持つモデルの図形や領域の図形などを設定または取得したいときは、設定対象または取得対象の図形に対して、図形データを配列で指定します。

9-6-1 図形データの構造一覧

図形データの構造は以下のとおりです。

配列要素	説明	
figure(0)	図形データヘッダ情報	図形データのヘッダ情報です。図形の個数と図形データサイズの情報が含まれます。 上位 16 ビット：図形の個数 下位 16 ビット：図形データサイズのバイト数（figure 配列長 × 4） 図形データヘッダ情報 = 図形データサイズのバイト数 + 図形データ個数 × 65536 図形の個数：図形データに含まれる図形の個数を設定します。通常は 1 を設定してください。複数の図形を組み合わせる場合は、2 以上の組み合わせる図形の個数を設定してください。 図形データサイズのバイト数：図形データサイズをバイト数に換算した値を設定します。配列要素数を 4 倍した値を設定してください。 （例）長方形 1 個の場合 配列長 = 5、図形データ個数 = 1 図形データサイズのバイト数 = 5 × 4 = 20 図形データヘッダ情報 = 20 + 1 × 65536 = 65556
figure(1)	図形 0 種別情報	図形 0 データの種別情報です。描画モードと図形種別の情報が含まれます。 上位 16 ビット：描画モード 下位 16 ビット：図形種別 図形種別情報 = 図形種別 + 描画モード × 65536 描画モード：図形の描画が OR モードか NOT モードかを設定します。 通常は 0（OR モード）を設定してください。2 個以上の図形を使用して図形の一部を除外する場合は、2 個目以降の図形で 1（NOT モード）を設定してください。 図形種別：直線や長方形など当該図形の種別を設定します。 （例）長方形 1 個（描画モードが OR）の場合 図形種別 = 8、描画モード = 0 図形種別情報 = 8 + 0 × 65536 = 8
figure(2)	図形 0 データ	図形 0 の図形データです。図形種別によってサイズと内容は異なります。
figure(M)	図形 1 種別情報	図形 1 データの種別情報です。
figure(M+1)	図形 1 データ	図形 1 の図形データです。図形種別によってサイズと内容は異なります。
:	:	:
figure(N*M)	図形 N 種別情報	図形 N データの種別情報です。
figure(N*M+1)	図形 N データ	図形 N の図形データです。図形種別によってサイズと内容は異なります。

9-6-2 図形データの配列要素一覧

図形ごとの配列要素と対応する設定項目は以下のとおりです。表中では、図形の個数が 1 の場合について記載しています。

図形	図形種別	配列要素	説明
点	1	figure(0)	図形データヘッダ情報
		figure(1)	図形種別情報
		figure(2)	座標 X
		figure(3)	座標 Y
直線	2	figure(0)	図形データヘッダ情報
		figure(1)	図形種別情報
		figure(2)	第 1 点 X 座標
		figure(3)	第 1 点 Y 座標
		figure(4)	第 2 点 X 座標
		figure(5)	第 2 点 Y 座標
幅付き直線	4	figure(0)	図形データヘッダ情報
		figure(1)	図形種別情報
		figure(2)	第 1 点 X 座標
		figure(3)	第 1 点 Y 座標
		figure(4)	第 2 点 X 座標
		figure(5)	第 2 点 Y 座標
		figure(6)	幅
長方形	8	figure(0)	図形データヘッダ情報
		figure(1)	図形種別情報
		figure(2)	左上点 X 座標
		figure(3)	左上点 Y 座標
		figure(4)	右下点 X 座標
		figure(5)	右下点 Y 座標
楕円	16	figure(0)	図形データヘッダ情報
		figure(1)	図形種別情報
		figure(2)	中心点 X 座標
		figure(3)	中心点 Y 座標
		figure(4)	X 方向半径
		figure(5)	Y 方向半径
円	32	figure(0)	図形データヘッダ情報
		figure(1)	図形種別情報
		figure(2)	中心点 X 座標
		figure(3)	中心点 Y 座標
		figure(4)	半径
幅付き円	64	figure(0)	図形データヘッダ情報
		figure(1)	図形種別情報
		figure(2)	中心点 X 座標
		figure(3)	中心点 Y 座標
		figure(4)	半径
		figure(5)	幅

図形	図形種別	配列要素	説明
円弧	128	figure(0)	図形データヘッダ情報
		figure(1)	図形種別情報
		figure(2)	中心点 X 座標
		figure(3)	中心点 Y 座標
		figure(4)	半径
		figure(5)	開始角度
		figure(6)	終了角度
幅付き円弧	256	figure(0)	図形データヘッダ情報
		figure(1)	図形種別情報
		figure(2)	中心点 X 座標
		figure(3)	中心点 Y 座標
		figure(4)	半径
		figure(5)	開始角度
		figure(6)	終了角度
		figure(7)	幅
多角形	512	figure(0)	図形データヘッダ情報
		figure(1)	図形種別情報
		figure(2)	頂点の数
		figure(3)	第 1 点 X 座標
		figure(4)	第 1 点 Y 座標
		figure(5)	第 2 点 X 座標
		figure(6)	第 2 点 Y 座標
		:	:
		figure(19)	第 9 点 X 座標
		figure(20)	第 9 点 Y 座標
		figure(21)	第 10 点 X 座標
		figure(22)	第 10 点 Y 座標

9-7 図形番号一覧

処理ユニットが持つモデルの図形や領域の図形などを設定または取得したいときは、設定対象または取得対象の図形を番号で指定します。処理項目ごとの図形番号と対応する図形は以下のとおりです。

処理項目	図形番号	説明
サーチ	0	モデル登録図形（モデル登録）
	1	計測領域図形（領域設定）
サーチⅡ	0	モデル登録図形（モデル登録）
	1	計測領域図形（領域設定）
フレキシブルサーチ	0	モデル 0 のモデル登録図形（モデル登録）
	1	モデル 1 のモデル登録図形（モデル登録）
	2	モデル 2 のモデル登録図形（モデル登録）
	3	モデル 3 のモデル登録図形（モデル登録）
	4	モデル 4 のモデル登録図形（モデル登録）
	5	モデル 5 のモデル登録図形（モデル登録）
センシティブサーチ	0	モデル登録図形（モデル登録）
	1	計測領域図形（領域設定）
ECM サーチ	0	モデル登録図形（モデル登録）
	1	計測領域図形（領域設定）
	2	マスク登録図形（モデル登録）
	3	モデル登録図形（エラーモデル登録）
	4	マスク登録図形（エラーモデル登録）
EC 円サーチ	0	モデル登録図形（モデル登録）
	1	計測領域図形（領域設定）
形状サーチⅡ	0	モデル登録図形（モデル登録）
	1	計測領域図形（領域設定）
形状サーチⅢ	0	モデル登録図形（モデル登録）
	1	計測領域図形（領域設定）
EC コーナー	0	計測領域図形（領域設定）
EC クロス	0	計測領域図形（領域設定）
仕分け	0	インデックス 0、 モデル 0 のモデル登録図形（モデル登録）
	1	インデックス 0、 モデル 1 のモデル登録図形（モデル登録）
	2	インデックス 0、 モデル 2 のモデル登録図形（モデル登録）
	3	インデックス 0、 モデル 3 のモデル登録図形（モデル登録）
	4	インデックス 0、 モデル 4 のモデル登録図形（モデル登録）
	5	インデックス 1、 モデル 0 のモデル登録図形（モデル登録）
	6	インデックス 1、 モデル 1 のモデル登録図形（モデル登録）
	:	:
	178	インデックス 35、 モデル 3 のモデル登録図形（モデル登録）
	179	インデックス 35、 モデル 4 のモデル登録図形（モデル登録）
	180	計測領域図形（領域設定）
	181	インデックス 36、 モデル 0 のモデル登録図形（モデル登録）
	182	インデックス 36、 モデル 1 のモデル登録図形（モデル登録）
	:	:
	999	インデックス 199、 モデル 3 のモデル登録図形（モデル登録）
	1000	インデックス 199、 モデル 4 のモデル登録図形（モデル登録）
エッジ位置	0	計測領域図形（領域設定）
エッジ本数	0	計測領域図形（領域設定）
スキャンエッジ位置	0	計測領域図形（領域設定）
スキャンエッジ幅	0	計測領域図形（領域設定）

処理項目	図形番号	説明
円形スキャンエッジ位置	0	計測領域図形（領域設定）
円形スキャンエッジ幅	0	計測領域図形（領域設定）
交点座標	0	直線 0 の計測領域図形（領域設定）
	2	直線 1 の計測領域図形（領域設定）
色平均・偏差	0	計測領域図形（領域設定）
面積重心	0	計測領域図形（領域設定）
ラベリング	0	計測領域図形（領域設定）
ラベルデータ	-	図形なし
キズ汚れ	0	計測領域図形（領域設定）
高精度キズ汚れ	0	計測領域図形（領域設定）
ファインマッチング	0	モデル登録図形（モデル登録）
汎用文字検査	0	計測領域図形（領域設定）
日付照合	-	図形なし
モデル辞書	0	インデックス 0、 モデル 0 のモデル登録図形（モデル登録）
	1	インデックス 0、 モデル 1 のモデル登録図形（モデル登録）
	2	インデックス 0、 モデル 2 のモデル登録図形（モデル登録）
	3	インデックス 0、 モデル 3 のモデル登録図形（モデル登録）
	4	インデックス 0、 モデル 4 のモデル登録図形（モデル登録）
	5	インデックス 1、
	6	インデックス 1、 モデル 1 のモデル登録図形（モデル登録）
	:	:
	178	インデックス 35、 モデル 3 のモデル登録図形（モデル登録）
	179	インデックス 35、 モデル 4 のモデル登録図形（モデル登録）
	180	計測領域図形（領域設定）
2 次元コード	0	計測領域図形（領域設定）
2 次元コードⅡ	0	計測領域図形（領域設定）
バーコード	0	計測領域図形（領域設定）
円形角度取得	0	計測領域図形（領域設定）
塗布剤途切れ検査	-	図形なし
AI ファインマッチング	-	図形なし
位置ずれ修正	0	計測領域図形（領域設定）
計測前処理	0	計測領域図形（領域設定）
背景カット	0	計測領域図形（領域設定）
明度補正フィルタ	0	計測領域図形（領域設定）
カラーグレーフィルタ	0	計測領域図形（領域設定）
色抽出フィルタ	0	計測領域図形（領域設定）
色ムラカット	0	計測領域図形（領域設定）
縞模様カットⅡ	0	計測領域図形（領域設定）
円展開	0	計測領域図形（領域設定）
台形歪み補正	0	計測領域図形（領域設定）
外部機器シミュレータ	-	図形なし
差分抽出	0	モデル登録図形（モデル登録）
高機能前処理	0	計測領域図形（領域設定）
パノラマ	-	図形なし
トリミング&リサイズ	0	計測領域図形（トリミング）
画像連結	-	図形なし

9-8 モデル番号一覧

処理ユニットのモデルを再登録したいときは、再登録対象のモデルを番号で指定します。処理項目ごとのモデル番号と対応するモデルは以下のとおりです。

処理項目	モデル番号	説明
サーチ	0	サーチモデル
サーチⅡ	0	サーチモデル
フレキシブルサーチ	0	インデックス 0、 モデル 0 のサーチモデル
	1	インデックス 0、 モデル 1 のサーチモデル
	2	インデックス 0、 モデル 2 のサーチモデル
	3	インデックス 0、 モデル 3 のサーチモデル
	4	インデックス 0、 モデル 4 のサーチモデル
センシティブサーチ	0	サーチモデル
ECM サーチ	0	サーチモデル
EC 円サーチ	0	サーチモデル
形状サーチⅡ	0	サーチモデル
形状サーチⅢ	0	サーチモデル
EC コーナー	0	基準位置
EC クロス	0	基準位置
仕分け	0	インデックス 0、 モデル 0 のサーチモデル
	1	インデックス 0、 モデル 1 のサーチモデル
	2	インデックス 0、 モデル 2 のサーチモデル
	3	インデックス 0、 モデル 3 のサーチモデル
	4	インデックス 0、 モデル 4 のサーチモデル
	5	インデックス 1、 モデル 0 のサーチモデル
	6	インデックス 1、 モデル 1 のサーチモデル
	:	:
	178	インデックス 35、 モデル 3 のサーチモデル
	179	インデックス 35、 モデル 4 のサーチモデル
	180	インデックス 36、 モデル 0 のサーチモデル
	181	インデックス 36、 モデル 1 のサーチモデル
	:	:
	998	インデックス 199、 モデル 3 のサーチモデル
	999	インデックス 199、 モデル 4 のサーチモデル
エッジ位置	0	エッジモデル
エッジ本数	0	エッジモデル
スキャンエッジ位置	0	エッジモデル
スキャンエッジ幅	0	エッジモデル
円形スキャンエッジ位置	0	エッジモデル
円形スキャンエッジ幅	0	エッジモデル
交点座標	0	エッジモデル
色平均・偏差	0	色平均・偏差モデル
面積重心	0	面積重心モデル
ラベリング	0	ラベリングモデル
ラベルデータ	-	モデルなし
キズ汚れ	0	キズ汚れモデル
高精度キズ汚れ	0	高精度キズ汚れモデル
ファインマッチング	0	ファインマッチングモデル
汎用文字検査	-	モデルなし
日付照合	-	モデルなし

処理項目	モデル番号	説明
モデル辞書	0	インデックス 0、 モデル 0 のサーチモデル
	1	インデックス 0、 モデル 1 のサーチモデル
	2	インデックス 0、 モデル 2 のサーチモデル
	3	インデックス 0、 モデル 3 のサーチモデル
	4	インデックス 0、 モデル 4 のサーチモデル
	5	インデックス 1、 モデル 0 のサーチモデル
	6	インデックス 1、 モデル 1 のサーチモデル
	:	:
	178	インデックス 35、 モデル 3 のサーチモデル
	179	インデックス 35、 モデル 4 のサーチモデル
2 次元コード	-	モデルなし
2 次元コード Ⅱ	-	モデルなし
バーコード	-	モデルなし
円形角度取得	-	モデルなし
塗布剤途切れ検査	-	モデルなし
AI ファインマッチング	-	モデルなし

9-9 画像番号一覧

処理ユニットが持つ画像データにアクセスしたいときは、アクセス対象の画像を番号で指定します。
処理項目ごとの画像番号と対応する画像は以下のとおりです。

処理項目	画像番号	説明
サーチ	-	画像なし
サーチⅡ	-	画像なし
フレキシブルサーチ	-	画像なし
センシティブサーチ	-	画像なし
ECM サーチ	-	画像なし
EC 円サーチ	-	画像なし
形状サーチⅡ	-	画像なし
形状サーチⅢ	-	画像なし
EC コーナー	-	画像なし
EC クロス	-	画像なし
仕分け	-	画像なし
エッジ位置	-	画像なし
エッジ本数	-	画像なし
スキャンエッジ位置	-	画像なし
スキャンエッジ幅	-	画像なし
円形スキャンエッジ位置	-	画像なし
円形スキャンエッジ幅	-	画像なし
交点座標	-	画像なし
色平均・偏差	-	画像なし
面積重心	-	画像なし
ラベリング	-	画像なし
ラベルデータ	-	画像なし
キズ汚れ	-	画像なし
高精度キズ汚れ	-	画像なし
ファインマッチング	-	画像なし
汎用文字検査	-	画像なし
日付照合	-	画像なし
モデル辞書	-	画像なし
2 次元コード	-	画像なし
2 次元コードⅡ	-	画像なし
バーコード	-	画像なし
OCR	-	画像なし
OCR ユーザ辞書	-	画像なし
円形角度取得	-	画像なし
塗布剤途切れ検査	-	画像なし
AI ファインマッチング	0	二値差分画像
	1	濃淡差分画像
	2	良品復元画像
	3	ラベリング二値画像
カメラ画像入力	0	カメラ画像
カメラ画像入力 FH	0	出力画像 0
	:	:
	7	出力画像 7
カメラ画像入力 HDR	0	カメラ画像
カメラ画像入力 HDR Lite	0	カメラ画像

処理項目	画像番号	説明
照度差ステレオ画像入力	0	出力画像
	1	形状画像 1
	2	形状画像 2
	3	形状画像 3
	4	テクスチャ画像
	16	撮像画像 1 枚目
	:	:
	23	撮像画像 7 枚目
カメラ切替	0	カメラ画像
計測画像切替	-	画像なし
位置ずれ修正	0	位置ずれ修正画像
計測前処理	0	前処理画像
背景カット	0	背景カット画像
明度補正フィルタ	0	明度補正画像
カラーグレーフィルタ	0	カラーグレー画像
色抽出フィルタ	0	色抽出画像
色ムラカット	0	色ムラカット画像
縞模様カットⅡ	0	縞模様カット画像
円展開	0	円展開画像
台形歪み補正	0	台形歪み補正画像
外部機器シミュレータ	0	軸移動後画像
差分抽出	0	差分抽出画像
高機能前処理	0	出力画像 0
	1	出力画像 1
	2	出力画像 2
	3	出力画像 3
パノラマ	0	パノラマ画像
トリミング&リサイズ	0	出力画像
画像連結	0	連結後画像
ユニットマクロ	0	処理ユニット画像 0
	1	処理ユニット画像 1
	:	:
	30	処理ユニット画像 30
	31	処理ユニット画像 31
ユニット演算マクロ	-	画像なし
演算	-	画像なし
近似直線	-	画像なし
近似円	-	画像なし
高精度キャリブレーション	0	補正後画像
ユーザデータ	-	画像なし
処理ユニットデータ設定	-	画像なし
処理ユニットデータ取得	-	画像なし
処理ユニット図形設定	-	画像なし
処理ユニット図形取得	-	画像なし
トレンドモニタ	-	画像なし
画像ロギング	-	画像なし
画像変換ロギング	-	画像なし
データロギング	-	画像なし
経過時間	-	画像なし
ウェイト	-	画像なし
フォーカス	-	画像なし

処理項目	画像番号	説明
アイリス	-	画像なし
並列化	-	画像なし
並列化タスク	-	画像なし
統計処理	-	画像なし
キャリブレーション参照	0	補正後画像
位置角度演算	-	画像なし
ステージデータ	-	画像なし
ロボットデータ	-	画像なし
画像マスタキャリブレーション	-	画像なし
PLC マスタキャリブレーション	-	画像なし
位置角度変換	-	画像なし
軸移動量演算	-	画像なし
多点軸移動量演算	-	画像なし
検出点	-	画像なし
カメラキャリブレーション	-	画像なし
データ保存	-	画像なし
条件分岐	-	画像なし
計測終了	-	画像なし
入力条件分岐	-	画像なし
無手順フロー制御	-	画像なし
PLC リンクフロー制御	-	画像なし
パラレルフロー制御	-	画像なし
Fieldbus フロー制御	-	画像なし
選択分岐	-	画像なし
シリアルデータ出力	-	画像なし
パラレルデータ出力	-	画像なし
パラレル判定出力	-	画像なし
Fieldbus データ出力	-	画像なし
結果表示	-	画像なし
画像ファイル表示	0	ファイル表示画像 0
	1	ファイル表示画像 1
	2	ファイル表示画像 2
	3	ファイル表示画像 3
最新 NG 画像表示	0	NG 画像 0
	1	NG 画像 1
	2	NG 画像 2
	3	NG 画像 3

9-10 サブ画像番号一覧

処理ユニットには、計測画像以外に処理中や処理結果の画像を、サブ画像として画像ウィンドウに表示できるものがあります。サブ画像を表示するときは、アクセス対象のサブ画像を番号で指定します。処理項目ごとのサブ画像番号と対応するサブ画像は以下のとおりです。

処理項目	画像番号	説明
サーチ	0	計測画像
サーチⅡ	0	計測画像
フレキシブルサーチ	0	計測画像
センシティブサーチ	0	計測画像
ECM サーチ	0	計測画像
	1	計測画像に一致したモデルを重ねた画像 緑：一致したモデル点 赤：一致しなかったモデル点
EC 円サーチ	0	計測画像
形状サーチⅡ	0	計測画像
形状サーチⅢ	0	計測画像
	1	計測画像に粗サーチモデルを重ねて表示
	2	エッジ画像
	3	エッジ画像に粗サーチモデルを重ねて表示
EC コーナー	0	計測画像
EC クロス	0	計測画像
仕分け	0	計測画像
エッジ位置	0	計測画像
	1	プロファイル画像
エッジ本数	0	計測画像
	1	プロファイル画像
スキャンエッジ位置	0	計測画像
	1	計測画像にスキャン領域を重ねて表示
スキャンエッジ幅	0	計測画像
	1	計測画像にスキャン領域を重ねて表示
円形スキャンエッジ位置	0	計測画像
	1	分割領域ごとのエッジ位置表示画像
円形スキャンエッジ幅	0	計測画像
	1	分割領域ごとのエッジ位置表示画像
交点座標	0	計測画像
色平均・偏差	0	計測画像
	1	マスク適用後画像
面積重心	0	計測画像
	1	色抽出画像（カラー画像を計測する場合） 2 値画像（モノクロ画像を計測する場合）
ラベリング	0	計測画像
	1	色抽出画像（カラー画像を計測する場合） 2 値画像（モノクロ画像を計測する場合）
ラベルデータ	0	計測画像
キズ汚れ	0	計測画像
	1	マスク適用後欠陥プロファイル画像（面積計測ありの場合）
高精度キズ汚れ	0	計測画像
	1	マスク適用後欠陥プロファイル画像（面積計測ありの場合） 欠陥領域画像（面積計測なしの場合）

処理項目	画像番号	説明
ファインマッチング	0	計測画像
	1	差分画像
汎用文字検査	0	計測画像
日付照合	0	計測画像
モデル辞書	0	計測画像
2 次元コード	0	計測画像
2 次元コードⅡ	0	計測画像
バーコード	0	計測画像
OCR	0	計測画像
OCR ユーザ辞書	0	計測画像
円形角度取得	0	計測画像
塗布剤途切れ検査	0	計測画像
	1	色抽出画像（カラー画像を計測する場合） 2 値画像（モノクロ画像を計測する場合）
	3	基準経路表示画像
	4	2 値画像
AI ファインマッチング	0	計測画像＋すべての欠陥位置、ラベル番号表示
	1	二値差分画像＋すべての欠陥位置、ラベル番号表示
	2	濃淡差分画像＋すべての欠陥位置、ラベル番号表示
	3	良品復元画像＋すべての欠陥位置、ラベル番号表示
	4	ラベリング二値画像＋すべての欠陥位置、ラベル番号表示
	5	欠陥重畳画像＋すべての欠陥位置、ラベル番号表示
	20	計測画像＋欠陥位置の円表示
	21	二値差分画像＋欠陥位置の円表示
	22	濃淡差分画像＋欠陥位置の円表示
	23	良品復元画像＋欠陥位置の円表示
	24	ラベリング二値画像＋欠陥位置の円表示
	25	欠陥重畳画像＋欠陥位置の円表示
	30	計測画像＋すべての欠陥位置の円表示
	31	二値差分画像＋すべての欠陥位置の円表示
	32	濃淡差分画像＋すべての欠陥位置の円表示
	33	良品復元画像＋すべての欠陥位置の円表示
	34	ラベリング二値画像＋すべての欠陥位置の円表示
	35	欠陥重畳画像＋すべての欠陥位置の円表示
カメラ画像入力	0	カメラ画像
カメラ画像入力 FH	0	カメラ画像
カメラ画像入力 HDR	0	カメラ画像
カメラ画像入力 HDR Lite	0	カメラ画像
カメラ切替	0	カメラ画像
計測画像切替	0	切替後画像
	1	切替前画像
位置ずれ修正	0	位置ずれ修正前画像
	1	位置ずれ修正後画像
計測前処理	0	前処理画像
背景カット	0	背景カット画像
明度補正フィルタ	0	明度補正画像
カラーグレーフィルタ	0	カラーグレー画像
色抽出フィルタ	0	色抽出画像
	1	計測画像
色ムラカット	0	色ムラカット画像
縞模様カットⅡ	0	縞模様カット画像

処理項目	画像番号	説明
円展開	0	円展開画像
	1	計測画像
台形歪み補正	0	台形歪み補正画像
外部機器シミュレータ	0	計測画像
	1	計測前の画像
差分抽出	0	差分抽出画像
	1	計測画像
高機能前処理	0	高機能前処理画像
	1	出力画像 0
	2	出力画像 1
	3	出力画像 2
	4	出力画像 3
	5	計測画像
パノラマ	0	パノラマ画像
トリミング&リサイズ	0	変換後の画像
	1	計測画像
画像連結	0	連結後画像
	1	計測画像
ユニットマクロ	0～100	任意に定義可能
ユニット演算マクロ	0	計測画像
演算	0	計測画像
近似直線	0	計測画像
近似円	0	計測画像
高精度キャリブレーション	0	高精度キャリブレーション画像
ユーザデータ	0	計測画像
処理ユニットデータ設定	0	計測画像
処理ユニットデータ取得	0	計測画像
処理ユニット図形設定	0	計測画像
処理ユニット図形取得	0	計測画像
トレンドモニタ	0	トレンドグラフ表示
	1	ヒストグラム表示
画像ロギング	0	計測画像
画像変換ロギング	0	計測画像
データロギング	0	計測画像
経過時間	0	計測画像
ウェイト	0	計測画像
フォーカス	0	計測画像
アイリス	0	計測画像
並列化	0	計測画像
並列化タスク	0	計測画像
統計処理	0	設定データの「データ番号」に対応するグラフ表示
	1	データ 0 のデータグラフ表示
	2	データ 1 のデータグラフ表示
	3	データ 2 のデータグラフ表示
	4	データ 3 のデータグラフ表示
	5	データ 4 のデータグラフ表示
	6	データ 5 のデータグラフ表示
	7	データ 6 のデータグラフ表示
	8	データ 7 のデータグラフ表示
キャリブレーション参照	0	キャリブレーション参照画像
位置角度演算	0	計測画像

処理項目	画像番号	説明
ステージデータ	0	計測画像
ロボットデータ	0	計測画像
画像マスタキャリブレーション	0	計測画像＋キャリブレーション進捗表示
	1	計測画像
PLC マスタキャリブレーション	0	計測画像＋キャリブレーション進捗表示
	1	計測画像
位置角度変換	0	計測画像
軸移動量演算	0	計測画像
多点軸移動量演算	0	計測画像
検出点	0	計測画像
強制座標設定	0	計測画像
カメラキャリブレーション	0	計測画像
データ保存	0	計測画像
条件分岐	0	計測画像
計測終了	0	計測画像
入力条件分岐	0	計測画像
無手順フロー制御	0	計測画像
PLC リンクフロー制御	0	計測画像
パラレルフロー制御	0	計測画像
Fieldbus フロー制御	0	計測画像
選択分岐	0	計測画像
シリアルデータ出力	0	計測画像
パラレルデータ出力	0	計測画像
パラレル判定出力	0	計測画像
Fieldbus データ出力	0	計測画像
結果表示	0	結果表示画像
画像ファイル表示	0	画像 0
	1	画像 1
	2	画像 2
	3	画像 3
最新 NG 画像表示	0	最新 NG 画像
	1	1 件前 NG 画像（保存画像が 2 枚以上の場合） 最新 NG 画像（保存画像が 2 枚未満の場合）
	2	2 件前 NG 画像（保存画像が 3 枚以上の場合） 最新 NG 画像（保存画像が 3 枚未満の場合）
	3	3 件前 NG 画像（保存画像が 4 枚以上の場合） 最新 NG 画像（保存画像が 4 枚未満の場合）

I

索引

索引

Numerics

*CLEARMEASUREDATA....5-18
*MCRINIT....5-16
*MEASUREDISPG....5-17
*MEASUREDISPI....5-17
*MEASUREDISPT....5-17
*MEASUREINIT....5-16
*MEASUREPROC....5-17
*RENUMPROC....5-17

A

Abs....8-21
AbsolutePath$....8-22
AddGlobalData....8-23
AddSystemData....8-24
AND....8-25
ApplicationPath$....8-26
ApplicationVersion$....8-27
ApproximationCircle....8-28
ArrayDims....8-29
ArrayLen....8-30
Asc....8-31
AssignUnit....8-32
Atn....8-33

B

BusyOut....8-34

C

Call（構造化サブルーチン呼出）....8-35
Call（ユーザ定義関数実行）....8-36
ChangeScene....8-38
ChangeSceneGroup....8-39
CheckUnit....8-40
Chr$....8-41
ClearMeasureData....8-42
ClearScene....8-43
ClearSceneGroup....8-44
Close....8-45
CloseTextData....8-46
Cls....8-48
Cont....8-49
ContinuousMeasure....8-50
CopyMeasureImage....8-51
CopyScene....8-52
CopySceneGroup....8-53
CopyUnit....8-54
CopyUnitFigure....8-55
CopyUnitImage....8-56
CopyUnitModel....8-57
Cos....8-59
Crspoint....8-60

D

Date$....8-62
Debug....8-63
DebugPrint....8-64
DeleteProfileData....8-65
DeleteUnit....8-66
Dim....8-67
DisableUnit....8-68
DisplaySubNo....8-69
DisplayUnitNo....8-70
Do Loop While....8-71
Dposline....8-72
DrawArc....8-74
DrawArcW....8-75
DrawBox....8-77
DrawCircle....8-79
DrawCircleW....8-80
DrawCursor....8-82
DrawEllipse....8-83
DrawFigure....8-85
DrawFillImage....8-86
DrawJudgeText....8-87
DrawLine....8-88
DrawLineW....8-90
DrawMeasureImage....8-91
DrawPoint....8-92
DrawPolygon....8-93
DrawSearchFigure....8-95
DrawText....8-97
DrawTextG....8-99
DrawUnitImage....8-100
Dskf....8-102

E

ElapsedTime....8-103
Eof....8-104
Erase....8-105
Errcmnd$....8-106
Errno....8-107
ErrorOut....8-108
EtherCAT....9-26
EtherNetIP....9-28
ExecuteErrorProc....8-110
ExecuteImageLogging....8-112
ExitFzProcess....8-113
Exp....8-114

F

Fcopy....8-115
FigureType....8-116
Fix....8-117
For To Step Next....8-118

G

GetAll....8-120
GetGlobalData....8-121
GetImageSize....8-122
GetImageWindow....8-123
GetMeasureOut....8-125
GetPlcData....8-126
GetPollingState....8-127
GetPort....8-129
GetProfileData....8-130
GetSceneData....8-131
GetSceneGroupData....8-132
GetSystemData....8-133
GetText$....8-134
GetTextWindow....8-137
GetUnitData....8-138
GetUnitFigure....8-139
Gosub....8-140
Goto....8-141

H

Hex$....8-142

I

If Then Else....8-143
If Then Elseif Else Endlf....8-144
ImageFormat....8-146
ImageUpdate....8-147
Input$....8-150
Input#....8-149
InsertUnit....8-152
Int....8-153
IO モジュール一覧....9-25
Isfile....8-154
ItemCount....8-155
ItemIdent$....8-156
ItemInfo....8-157
ItemTitle$....8-158

J

JudgeOut....8-159

K

Kill....8-160

L

LanguageString$....8-161
LCase$....8-162
Left$....8-163
Len....8-164
Line Input#....8-165
List....8-167
LoadBackupData....8-168
LoadScene....8-169
LoadSceneGroup....8-170
LoadSystemData....8-171
LoadUnitData....8-172
Log....8-174
Lsqumeth....8-175

M

Macro(IO)>....6-5
Macro(SC)>....6-5
Macro(U**)>....6-5
Measure....8-176
MeasureDispG....8-177
MeasureDispI....8-178
MeasureDispT....8-179
MeasureId$....8-180
MeasureProc....8-181
MeasureStart....8-182
MeasureStop....8-183
MemoryInfo....8-184
MessageBox....8-185
Mid$....8-186
Mkdir....8-188
MOD....8-189
MoveUnit....8-190

N

NOT....8-191

O

Open For Append As#....8-192
Open For Input As#....8-193
Open For Output As#....8-194
OpenTextData....8-196
Option Explicit....8-198
OR....8-199

P

Parallello....9-34
Piece$....8-200
PLC リンク(JEPMC MP)....9-52
PLC リンク(MELSEC QnU/Q/QnAS)....9-38
PLC リンク(MELSEC QnU/Q/QnAS)(TCP)....9-58
PLC リンク(MELSEC QnU/Q/QnAS)(UDP)....9-49
PLC リンク(SYSMAC CS/CJ/CP/One)....9-40
PLC リンク(SYSMAC CS/CJ/CP/One)(TCP)....9-61
PLC リンク(SYSMAC CS/CJ/CP/One)(UDP)....9-55
Print....8-202
Print#....8-203
PROFINET....9-31
PutAll....8-204
PutPort....8-205

R

RaiseErrorProcEvent....8-207
RaiseOptionEvent....8-209

ReadPlcMemory....8-210
ReceiveData....8-212
ReDim....8-214
RefreshImageWindow....8-215
RefreshJudgeWindow....8-216
RefreshTextWindow....8-217
RefreshTimeWindow....8-218
Rem....8-219
Remeasure....8-220
RenumUnitNo....8-221
RGB....8-222
Right$....8-223
Rmdir....8-224
RunOut....8-225

S

SaveBackupData....8-226
SaveData....8-227
SaveImage....8-228
SaveMeasureImage....8-229
SaveScene....8-231
SaveSceneGroup....8-232
SaveSystemData....8-233
SaveUnitData....8-234
SceneCount....8-235
SceneDescription$....8-236
SceneGroupCount....8-237
SceneGroupNo....8-238
SceneGroupTitle$....8-239
SceneMaker$....8-240
SceneNo....8-241
SceneTitle$....8-242
ScreenCapture....8-243
Select Case Case Else End Select....8-244
SendData....8-246
SendString....8-248
SerialNormal....9-36
SerialPlcLink....9-40
SerialPlcLinkM....9-38
SetDisplayUnitNo....8-249
SetDrawStyle....8-250
SetForegroundLine....8-252
SetGlobalData....8-253
SetImageWindow....8-254
SetMeasureImage....8-256
SetMeasureOut....8-257
SetPlcData....8-258
SetPollingState....8-260
SetProfileData....8-262
SetSceneData....8-263
SetSceneDescription....8-265
SetSceneGroupData....8-266
SetSceneGroupTitle....8-267
SetSceneMaker....8-268
SetSceneTitle....8-269
SetStop....8-270
SetSystemData....8-271
SetTextStyle....8-272
SetTextWindow....8-274
SetUnitData....8-275
SetUnitFigure....8-276
SetUnitJudge....8-277
SetUnitTitle....8-278
SetUserSubroutine....8-279
SetVar....8-281
Sin....8-283
Sqr....8-284
StartTimer....8-285
Stop....8-286
Str$....8-287
Str2$....8-288
SubList....8-292
Sub ～ End Sub....8-290
SystemInfo....8-293
SystemReset....8-294

T

Tan....8-295
TcpClient....9-43
TcpNormal....9-45
TcpPlcLink....9-61
TcpPlcLinkM....9-58
TestMeasure....8-296
Time$....8-297
Timer....8-298
TotalJudge....8-300
TransformAngle....8-301
TransformArea....8-302
TransformDist....8-303
TransformLine....8-304
TransformXY....8-305
Try Catch End Try....8-307

U

UCase$....8-309
UdpNormal....9-47
UdpPlcLink....9-55
UdpPlcLinkM....9-49
UdpPlcLinkY....9-52
UDP 無手順通信....9-47
UnitCount....8-310
UnitData....8-311
UnitData$....8-312
UnitData2....8-313
UnitInfo....8-314
UnitItemIdent$....8-316
UnitJudge....8-317
UnitNo....8-318
UnitTitle$....8-319
Ut....8-320

V

Val....8-321
VarList....8-322
VarPop....8-323

VarPush....8-325
VarSave....8-327

W

Wait....8-328
WritePlcMemory....8-329

X

XOR....8-331

あ

アスタリスク(*)....4-3
アットマーク(@)....4-3
アポストロフィ(‘)....4-3
アンパサンド(&)....4-3

い

一時変数....4-7

え

エラー一覧....9-2
演算子....4-12

か

外部参照データ....5-2
画像データ....5-6
画像番号一覧....9-74
画像表示 (*MEASUREDISPI)....5-17
関係演算子....4-13
関係式....4-14
関数....4-14

き

基本構文....4-2
行....4-3
行番号....4-3

く

グラフィック表示 (*MEASUREDISPG)....5-17
グローバルデータ....5-7

け

計測受付状態....5-13
計測禁止状態....5-13
計測結果クリア (*CLEARMEASUREDATA)....5-18
計測実行 (*MEASUREPROC)....5-17
計測初期化 (*MEASUREINIT)....5-16
計測処理の排他制御....5-18
計測フロー更新 (*RENUMPROC)....5-17

こ

コロン(:)....4-3
コンマ(,)....4-3

さ

サブ画像番号一覧....9-77
サブルーチン....4-4
算術演算子....4-12
参照変数....4-9

し

式....4-14
システム状態監視コンソールウィンドウの見かた....3-6
システムデータ一覧....9-10
システム変数....4-11
システムデータ....5-8
実行文....4-3
シャープ(#)....4-3
16 進数....4-5
10 進数....4-5
シリアル無手順通信....9-36
シーン制御マクロツール....1-2
シーン制御マクロツールの設定画面の見かたと設定方法....3-12
シーン制御マクロでのプログラムのセーブとロード....3-25
シーン変数....4-11

す

数式....4-14
図形データ....5-3
図形データ一覧....9-67
図形番号一覧....9-70
スペース( )....4-3

せ

整数型....4-5, 4-6
セミコロン(;)....4-3

た

代入演算子....4-12
ダブルクォーテーション(")....4-3

ち

注釈コメント....4-4

つ

通信コマンドマクロツール....1-2
通信コマンドマクロツールの設定画面の見かたと設定方法....3-13
通信コマンドマクロでのプログラムのセーブとロード....3-25
通信処理の排他制御....5-19

て

定数....4-5

TDM エディタ..... 4-11
テキスト表示 (*MEASUREDISPT)..... 5-17
デバッグ機能の使い方について..... 6-2
デバッグモード..... 6-2
データのスコープ..... 5-9
データの保存領域..... 5-10

と

トラブルシューティングについて..... 7-2
ドル($)..... 4-3

に

2 進数..... 4-5

は

実数型..... 4-5, 4-6
配列変数..... 4-8
8 進数..... 4-5
パラレル I/O..... 9-34

ひ

非実行文..... 4-3

ふ

プログラム初期化 (*MCRINIT)..... 5-16
プログラム編集画面の見かた..... 3-2
文（ステートメント）..... 4-3

へ

変数..... 4-6

む

無手順（TCP クライアント）..... 9-43
無手順(TCP) 通信..... 9-45

も

文字..... 4-3
文字式..... 4-14
文字列型..... 4-5, 4-6
モデルデータ..... 5-5
モデル番号一覧..... 9-72

ゆ

ユニット演算マクロ処理項目..... 1-2
ユニット演算マクロ処理項目の設定画面の見かたと設定方法..... 3-7
ユニット演算マクロでのプログラムのセーブとロード. 3-24
ユニットマクロ処理項目..... 1-2
ユニットマクロ処理項目の設定画面の見かたと設定方法..... 3-20
ユニットマクロでのプログラムのセーブとロード..... 3-25
ユーザデータ..... 5-8

よ

予約一覧..... 9-5

ら

ラベル..... 4-3, 4-4

り

リリースモード..... 6-2

ろ

論理演算子..... 4-13
論理式..... 4-14

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

フリー通話 **0120-919-066**（クイック　オムロン）

携帯電話の場合、

055-982-5015（有料）をご利用ください。

受付時間：9:00〜17:00（土・日・12/31〜1/3を除く）

オムロンFAクイックチャット

www.fa.omron.co.jp/contact/tech/chat/

技術相談員にチャットでお問い合わせいただけます。（I-Webメンバーズ限定）

受付時間：平日9:00〜12:00 / 13:00〜17:00（土日祝日・年末年始・当社休業日を除く）

※受付時間、営業日は変更の可能性がございます。最新情報はリンク先をご確認ください。

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いただけます。緊急時のご購入にもご利用ください。 **www.fa.omron.co.jp**

●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。

オムロン商品のご用命は

Man. No. SDNB-721S　2026年9月現在

©OMRON Corporation 2016-2026 All Rights Reserved.
お断りなく仕様などを変更することがありますのでご了承ください