



ALGOSYSTEM

MULTIPROG 用 汎用機能

ファンクションブロック

リファレンスマニュアル

株式会社アルゴシステム

www.algosystem.co.jp

77KW10031A | 2019年7月 | 初版 | リリース |

目次

はじめに

| | |
|----------------|---|
| 1) ……お願いと注意 …… | 1 |
|----------------|---|

第1章 MULTIPROG 用 汎用機能ライブラリ

| | |
|------------------------|-----|
| 1-1 ·MULTIPROG とは····· | 1-1 |
|------------------------|-----|

第2章 ファンクションブロック

| | |
|-----------------------------|-----|
| 2-1 ·機能概要····· | 2-1 |
| 2-2 ·使用方法····· | 2-3 |
| 2-3 ·ファンクションブロックリファレンス····· | 2-7 |

第3章 付録

| | |
|----------------|-----|
| 3-1 ·参考文献····· | 3-1 |
|----------------|-----|

はじめに

この度は、アルゴシステム製品をお買い上げ頂きありがとうございます。

弊社製品を安全かつ正しく使用していただくために、お使いになる前に本書をお読みいただき、十分に理解していただくようお願い申し上げます。

1) お願いと注意

本書では、下記の方法について説明します。

- ・ PLC プログラミング用汎用機能ファンクションブロックライブラリの使用方法

MULTIPROG や PLC プログラミングについての詳細は省略させていただきます。

MULTIPROG および PLC プログラミングに関する資料および文献と併せて本書をお読みください。

第 1 章 MULTIPROG 用 汎用機能ライブラリ

本章では PHOENIX CONTACT 社製 MULTIPROG における汎用機能ライブラリについて、基本的な仕様、構成について説明します。

1-1 MULTIPROG とは

MULTIPROG とは、PHOENIX CONTACT 社が開発した、IEC に基づいて設計された PLC や従来からの PLC のための、標準的なプログラミングシステムです。

MULTIPROG は IEC61131-3 規格に基づいており、IEC の特徴を最大限含みます。

第2章 ファンクションブロック

本章では、ファンクションブロックについて説明します。

ファンクションブロックを使用する際のファームウェアライブラリ名称は「MP_AlCmn_SizeofArray」になります。

2-1 機能概要

汎用機能

| ファンクション ブロック名 | 機 能 |
|----------------------|---|
| AlCmn_SizeofArray | 渡された型や変数のメモリサイズを調べます |
| AlCmn_AlwaysON | 常に ON (True) を返します |
| AlCmn_AlwaysOFF | 常に OFF (False) を返します |
| AlCmn_TimeToValue | TIME 型データを日/時/分/秒/ミリ秒に変換します |
| AlCmn_ValueToTime | 日/時/分/秒/ミリ秒を TIME 型データに変換します |
| AlCmn_PulseCount | 指定した時間に入力されるパルスのカウントします |
| AlCmn_PulseOut | 指定した周期で指定したカウント数だけパルスを出力します |
| AlCmn_PWM | 指定した時間幅のパルスを指定した周期で出力します |
| AlCmn_SBIT_BYTE | BYTE 型データの N ビット目をセットします |
| AlCmn_SBIT_WORD | WORD 型データの N ビット目をセットします |
| AlCmn_SBIT_DWORD | DWORD 型データの N ビット目をセットします |
| AlCmn_RBIT_BYTE | BYTE 型データの N ビット目をリセットします |
| AlCmn_RBIT_WORD | WORD 型データの N ビット目をリセットします |
| AlCmn_RBIT_DWORD | DWORD 型データの N ビット目をリセットします |
| AlCmn_TBIT_BYTE | BYTE 型データの N ビット目をテスト (0 or 1 を判断) します |
| AlCmn_TBIT_WORD | WORD 型データの N ビット目をテスト (0 or 1 を判断) します |
| AlCmn_TBIT_DWORD | DWORD 型データの N ビット目をテスト (0 or 1 を判断) します |
| AlCmn_DECODE_BYTE | 指定したビットを ON した BYTE 型データを出力します |
| AlCmn_DECODE_WORD | 指定したビットを ON した WORD 型データを出力します |
| AlCmn_DECODE_DWORD | 指定したビットを ON した DWORD 型データを出力します |
| AlCmn_ENCODE_BYTE | BYTE 型データの内、最も下位の ON しているビット番号を出力します |
| AlCmn_ENCODE_WORD | WORD 型データの内、最も下位の ON しているビット番号を出力します |
| AlCmn_ENCODE_DWORD | DWORD 型データの内、最も下位の ON しているビット番号を出力します |
| AlCmn_BITCOUNT_BYTE | BYTE 型データの内、ON しているビットの個数を出力します |
| AlCmn_BITCOUNT_WORD | WORD 型データの内、ON しているビットの個数を出力します |
| AlCmn_BITCOUNT_DWORD | DWORD 型データの内、ON しているビットの個数を出力します |
| AlCmn_DBAND_INT | 指定した INT 型データを不感帯幅として、出力が変化します |
| AlCmn_DBAND_DINT | 指定した DINT 型データを不感帯幅として、出力が変化します |
| AlCmn_DBAND_REAL | 指定した REAL 型データを不感帯幅として、出力が変化します |
| AlCmn_BIAS_INT | 指定した INT 型データをバイアスとして、出力が変化します |
| AlCmn_BIAS_DINT | 指定した DINT 型データをバイアスとして、出力が変化します |
| AlCmn_BIAS_REAL | 指定した REAL 型データをバイアスとして、出力が変化します |

| ファンクション ブロック名 | 機 能 |
|---------------------|---|
| AI Cmn_PACK_WORD | BYTE 型データ 2 つを結合し、WORD 型データを出力します |
| AI Cmn_PACK_DWORD | BYTE 型データ 4 つを結合し、DWORD 型データを出力します |
| AI Cmn_UNPACK_WORD | WORD 型データを BYTE 型データに分離し、出力します |
| AI Cmn_UNPACK_DWORD | DWORD 型データを BYTE 型データに分離し、出力します |
| AI Cmn_SWAP_WORD | WORD 型データの上位バイトと下位バイトを入れ替え出力します |
| AI Cmn_SWAP_DWORD | DWORD 型データの上位バイトと下位バイトを入れ替え出力します |
| AI Cmn_MUX_INT | 複数の INT 型データの中から選択した 1 つの INT 型データを出力します |
| AI Cmn_MUX_UINT | 複数の UINT 型データの中から選択した 1 つの UINT 型データを出力します |
| AI Cmn_MUX_DINT | 複数の DINT 型データの中から選択した 1 つの DINT 型データを出力します |
| AI Cmn_MUX_UDINT | 複数の UDINT 型データの中から選択した 1 つの UDINT 型データを出力します |
| AI Cmn_MUX_REAL | 複数の REAL 型データの中から選択した 1 つの REAL 型データを出力します |
| AI Cmn_MUX_LREAL | 複数の LREAL 型データから 1 つの LREAL 型データを出力します |
| AI Cmn_MUX_STRING | 複数の STRING 型データから 1 つの STRING 型データを出力します |
| AI Cmn_MUX_TIME | 複数の TIME 型データから 1 つの TIME 型データを出力します |
| AI Cmn_HALM_WORD | 入力した WORD 型データが上限設定値を超える値の場合に警報を出力します |
| AI Cmn_HALM_DWORD | 入力した DWORD 型データが上限設定値を超える値の場合に警報を出力します |
| AI Cmn_LALM_WORD | 入力した WORD 型データが下限設定値未満の値の場合に警報を出力します |
| AI Cmn_LALM_DWORD | 入力した DWORD 型データが下限設定値未満の値の場合に警報を出力します |
| AI Cmn_HLALM_WORD | 入力した WORD 型データが下限設定値未満、または上限設定値を超える値の場合に警報を出力します |
| AI Cmn_HLALM_DWORD | 入力した DWORD 型データが下限設定値未満、または上限設定値を超える値の場合に警報を出力します |
| AI Cmn_TMSUM | システム起動時からの経過時間を出力します |

2-2 使用方法

MULTIPROG のプロジェクトでユニット毎のファンクションブロックを使用するためには、プロジェクト毎に登録が必要となります。

本項では登録方法について説明します。

ライブラリのインストールパスは「<MULTIPROG インストール先ディレクトリ>\%plc\FW_LIB」になります。

①MULTIPROG 画面の左ペインにある「ライブラリ」を選択します。

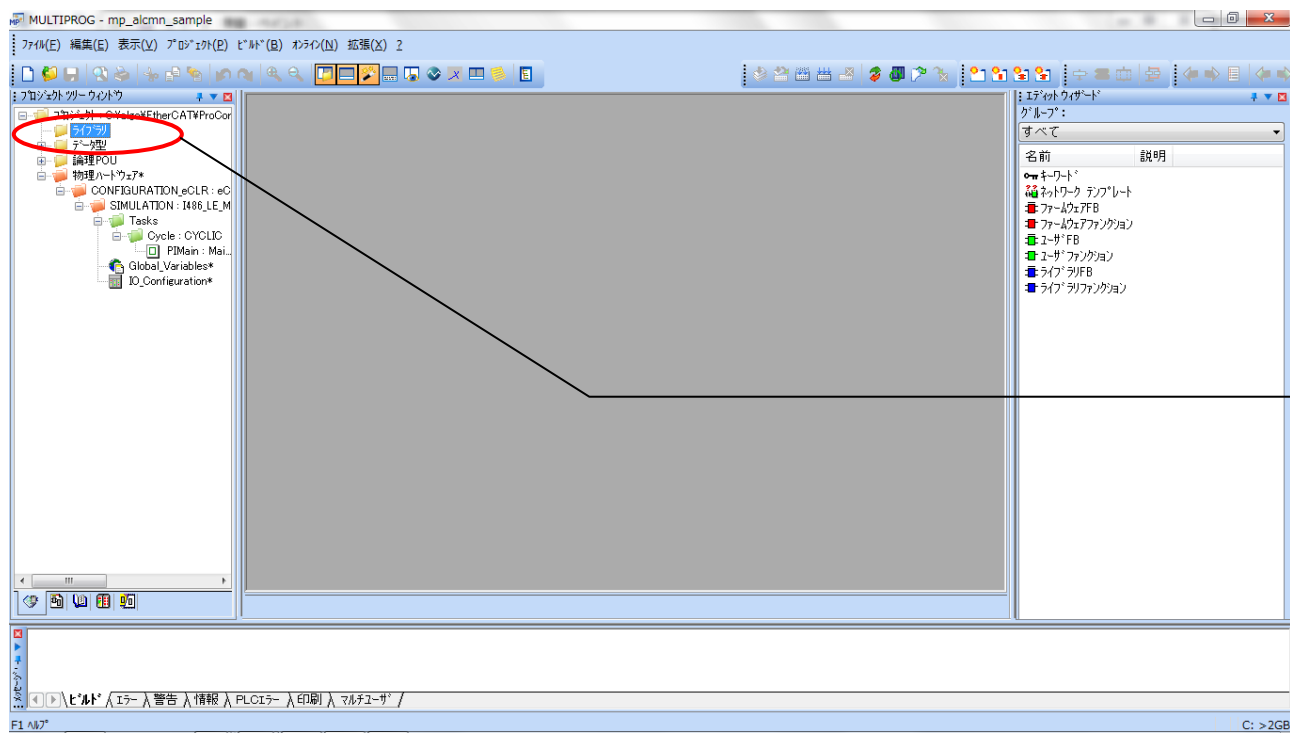


図 2-2-1. ライブラリ選択画面

② 「ライブラリ」を右クリックし「挿入(I)」→「ファームウェアライブラリ(F)」と選択します。

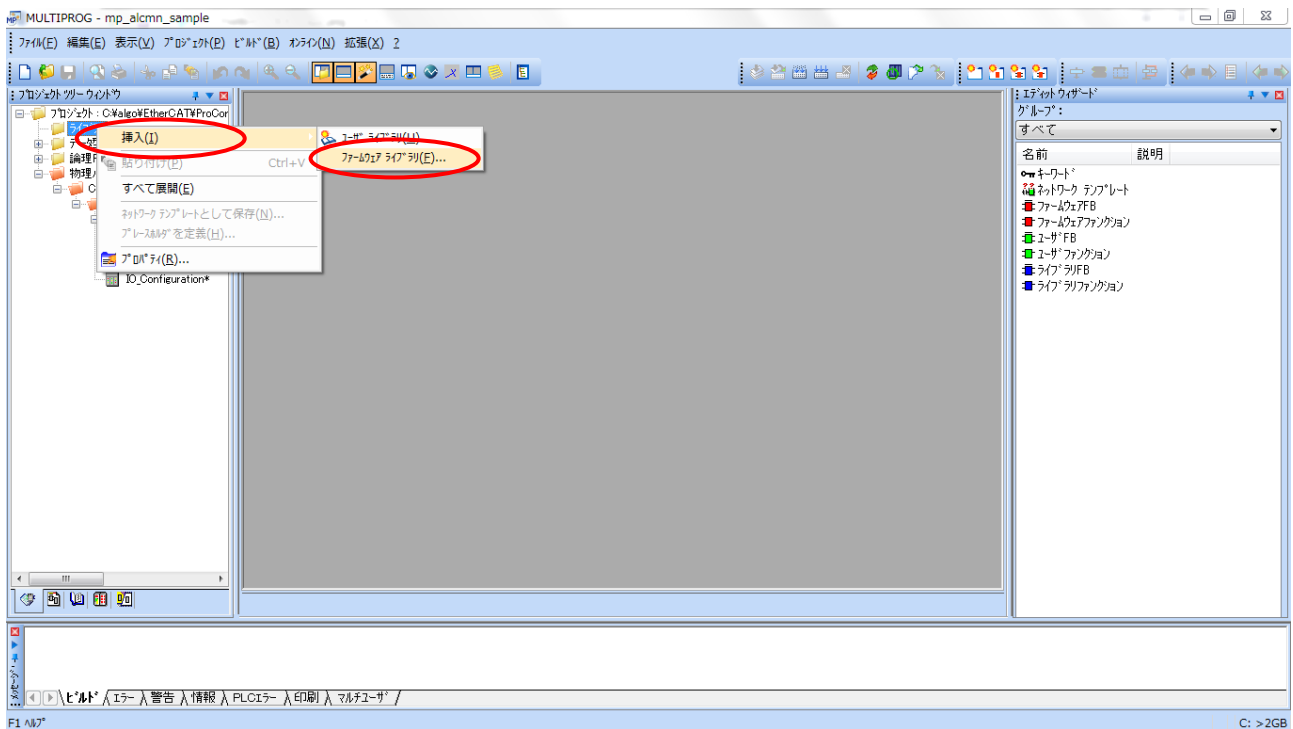


図 2-2-2. ライブラリ挿入画面①

- ③ 図 2-2-3 の画面が表示されますので、使用するファームウェアライブラリと同じ名前のディレクトリを選択してください。

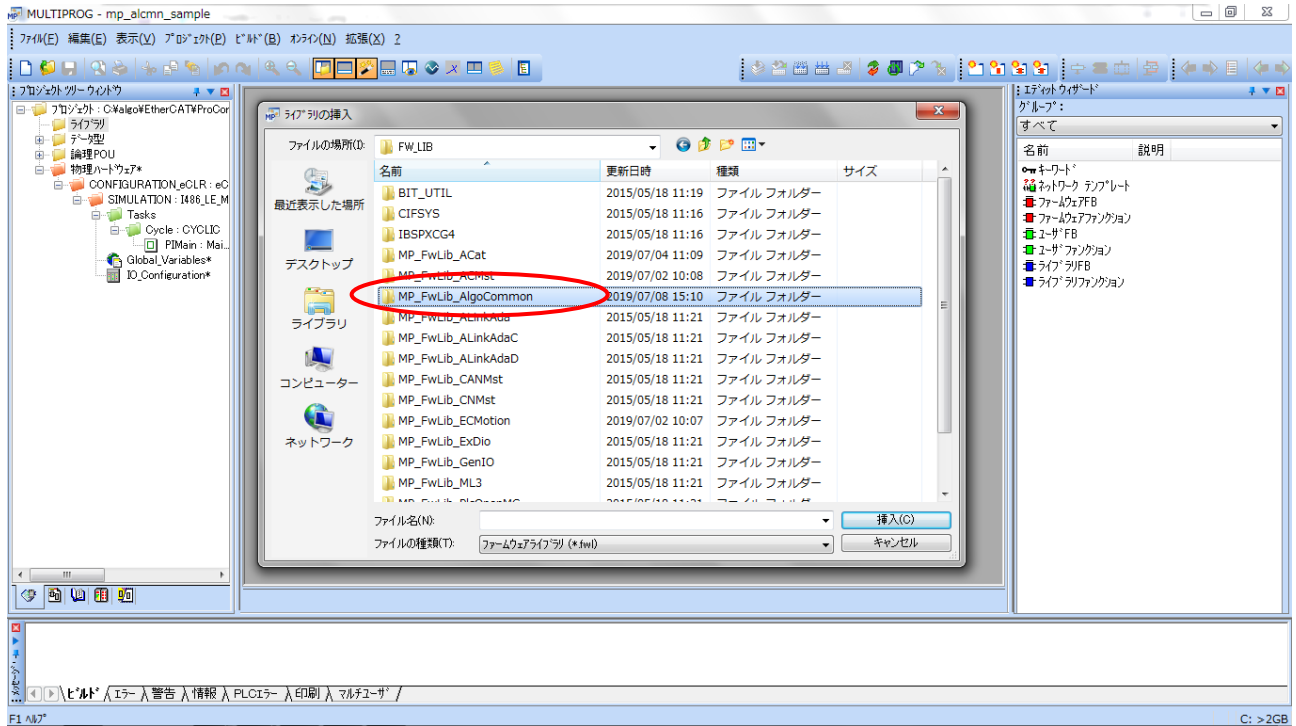


図 2-2-3. ライブラリ挿入画面②

- ④ 図 2-2-4 の画面が表示されますので、選択したディレクトリと同じ名前のファイルを選択してください。

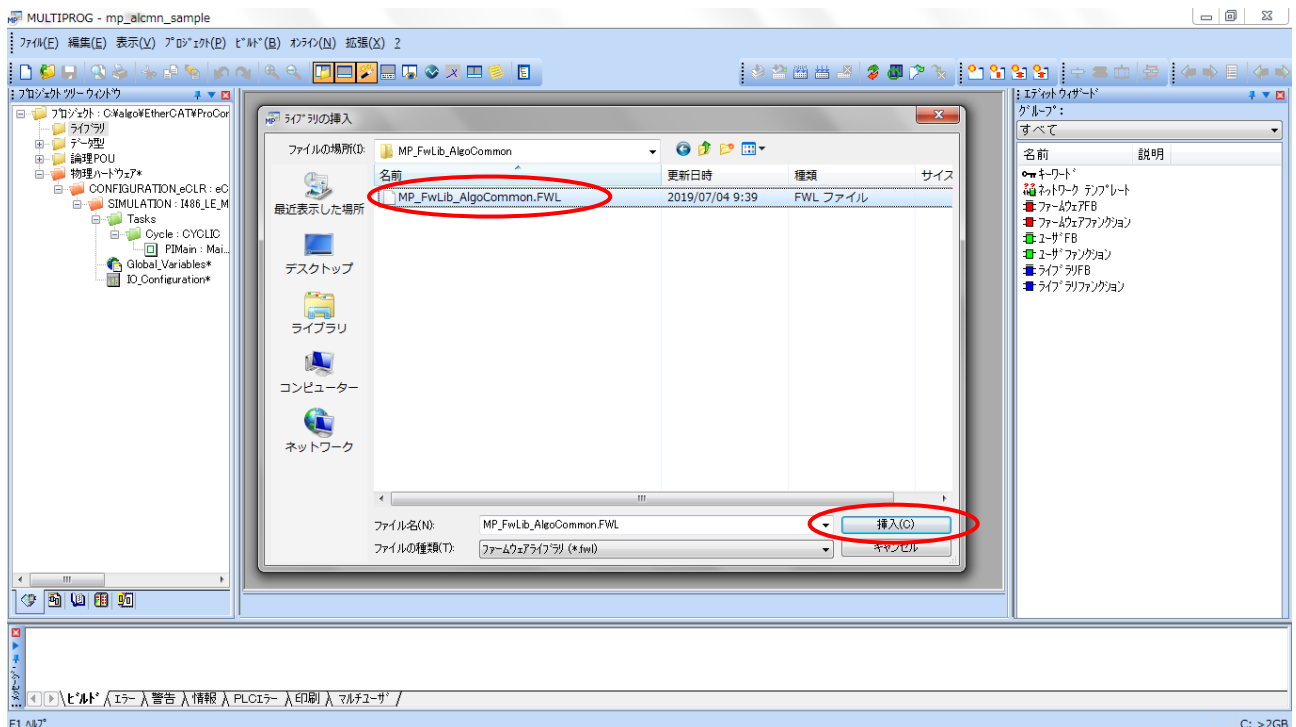


図 2-2-4. ライブラリ挿入画面③

- ④最後に「挿入(C)」ボタンを押すことで、登録が完了します。
ライブラリの項に選択したライブラリが追加されている事を確認してください。

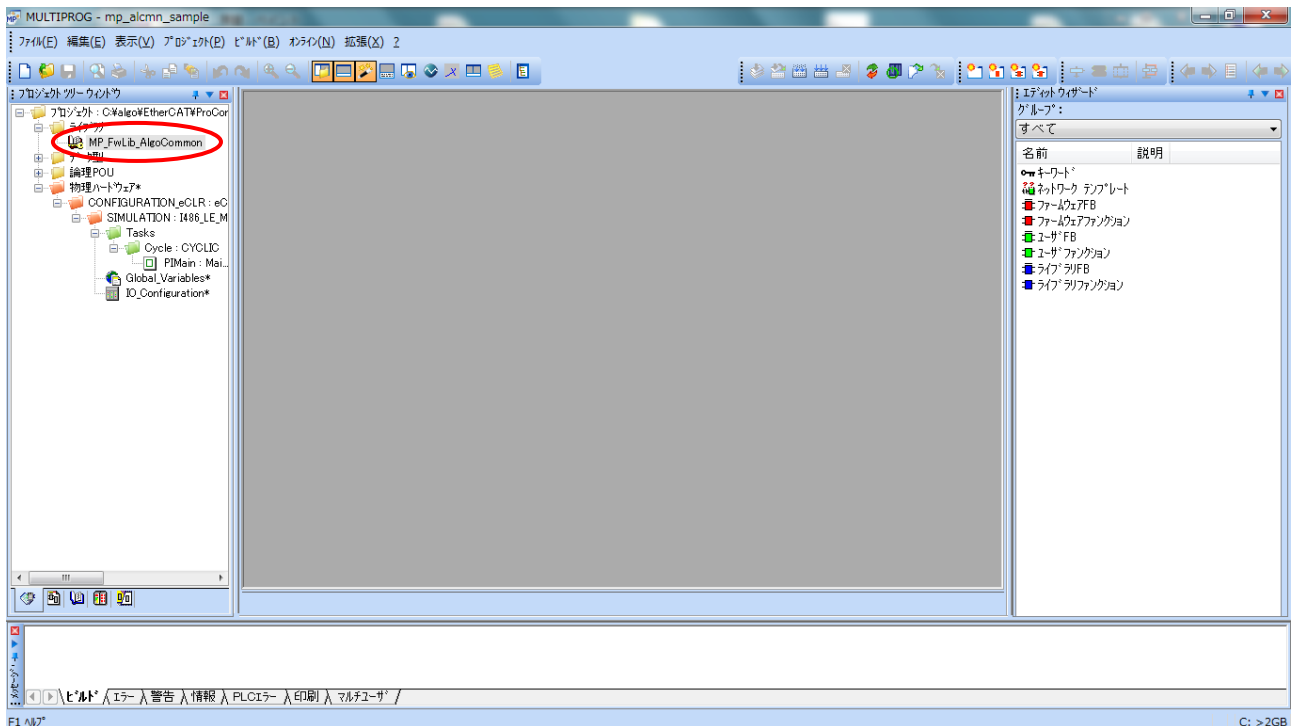


図 2-2-5. ライブラリ挿入画面④

2-3 ファンクションブロックリファレンス

本項では MULTIPROG 用として用意したファンクションブロックについて、ファンクションブロックに共通の入出力パラメータについて説明します。

各ファンクションブロックには処理を実行するための入力として「Enable」、処理の結果を知らせるための出力として「Done」「Error」を用意しています。全てのファンクションブロックに共通の制御を行っています。これらの制御について以下で説明します。

Enable

本パラメータを FALSE から TRUE への変化させることで処理を開始します。

本パラメータを FALSE にする事で応答待ちを停止し、ファンクションブロックの全ての出力パラメータを初期化します。

Done

本出力パラメータが TRUE になると処理の正常完了になります。

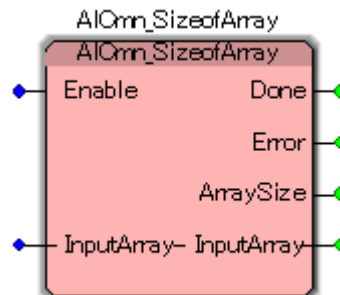
Error

ファンクションブロックが正常終了の場合、FALSE となります。

本出力パラメータが TRUE になった場合は、異常があります。

AIOmn_SizeofArray 関数

機能 構造体や型、変数のメモリサイズを調べます

書式

入力 BOOL Enable : アクションフラグ (0:停止, 1:実行)

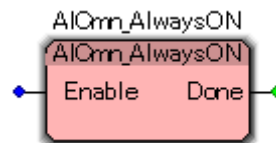
出力 BOOL Done : リターン (0:応答なし, 1:実行終了)
WORD ArraySize : メモリサイズ
BOOL Error : エラーフラグ (1:正常, 0:異常)

説明 InputArray で指定した構造体や型、変数のメモリサイズを調べます。
Done に TRUE が返った後、ArraySize に調べたメモリサイズが格納されます。

AIOmn_AlwaysON 関数

機能 常に ON を返します

書式



入力 BOOL Enable : アクションフラグ (0:停止, 1:実行)

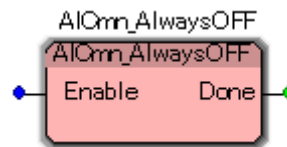
出力 BOOL Done : リターン (1:ON)

説明 Enable に関係なく、常に ON を返します。

AIOmn_AlwaysOFF 関数

機能 常に OFF を返します

書式



入力 BOOL Enable : アクションフラグ (0:停止, 1:実行)

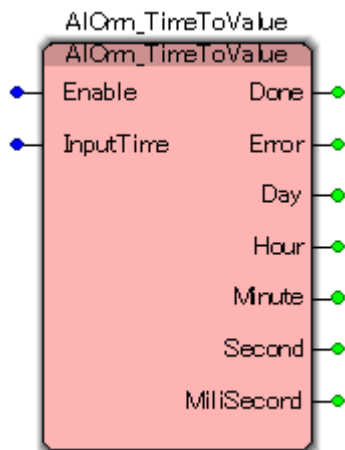
出力 BOOL Done : リターン (0:OFF)

説明 Enable に関係なく、常に OFF を返します。

AICmn_TimeToValue 関数

機能

TIME 型データを日/時/分/秒/ミリ秒に変換します

書式**入力**

BOOL Enable : アクションフラグ (0:停止, 1:実行)
 TIME InputTime : 変換前の継続時間

出力

BOOL Done : リターン (0:応答なし, 1:実行終了)
 BOOL Error : エラーフラグ (0:正常, 1:異常)
 UINT Day : 日 (0~49)
 UINT Hour : 時 (0~23)
 UINT Minute : 分 (0~59)
 UINT Second : 秒 (0~59)
 UINT MilliSecond : ミリ秒 (0~999)

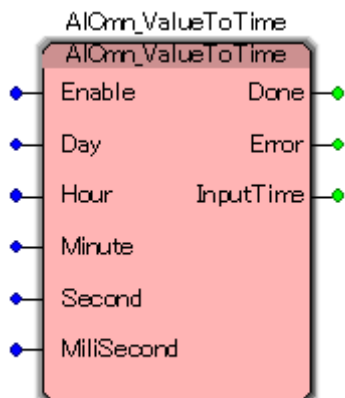
説明

本 FB は入力 Enable に TRUE を入力した時に変換を行います。
 Done に TRUE が返った後、Day/Hour/Minute/Second/MilliSecond に変換した
 継続時間が格納されます。
 Enable が TRUE から FALSE になった際、出力は保持されます。
 TIME 型が扱える範囲は 0~4, 294, 967, 295ms です。

AIOmn_ValueToTime 関数

機能

日/時/分/秒/ミリ秒を TIME 型データに変換します

書式**入力**

| | | |
|------|-------------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| UINT | Day | : 日 (0~49) |
| UINT | Hour | : 時 (0~23) |
| UINT | Minute | : 分 (0~59) |
| UINT | Second | : 秒 (0~59) |
| UINT | MilliSecond | : ミリ秒 (0~999) |

出力

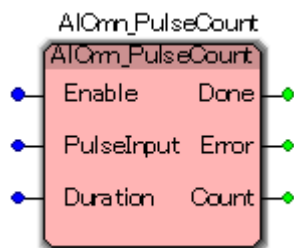
| | | |
|------|-----------|---------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| BOOL | Error | : エラーフラグ (0: 正常, 1: 異常) |
| TIME | InputTime | : 変換後の継続時間 |

説明

本 FB は入力 Enable に TRUE を入力した時に変換を行います。
Done に TRUE が返った後、InputTime に変換した継続時間が格納されます。
Enable が TRUE から FALSE になった際、出力は保持されます。
TIME 型が扱える範囲は 0~4, 294, 967, 295ms です。

AIOmn_PulseCount 関数

機能 指定した時間に入力されるパルスをカウントします

書式

入力

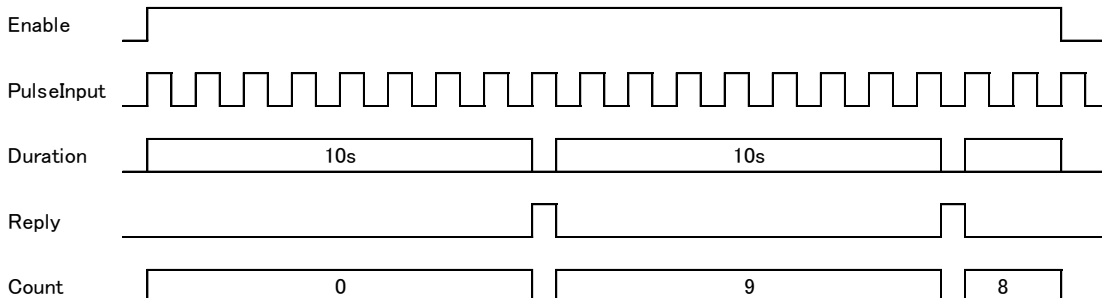
| | | |
|------|------------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| BOOL | PulseInput | : 入力パルス |
| UINT | Duration | : 指定時間 (秒) |

出力

| | | |
|------|-------|---------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| BOOL | Error | : エラーフラグ (0: 正常, 1: 異常) |
| BOOL | Count | : パルスカウント |

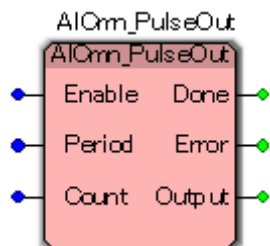
説明

本 FB は入力 Enable を ON すると Duration で指定した時間、PulseInput に入力されたパルスをカウントします。
 指定時間経過で、Done が ON し、Count にカウント数を格納します。
 Done が ON した後、まだ Enable が ON の場合、1 スキャン経過で Done が OFF し、再びカウントを開始します。
 Enable が ON から OFF した時、Count は保持されます。
 Enable を OFF から ON した時、Count はクリアされます。

動作

AIOmn_PulseOut 関数

機能 指定した周期で指定したカウント数だけパルスを出力します

書式

入力

| | | |
|------|--------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| UINT | Period | : パルス周期 (秒) |
| UINT | Count | : パルスカウント |

出力

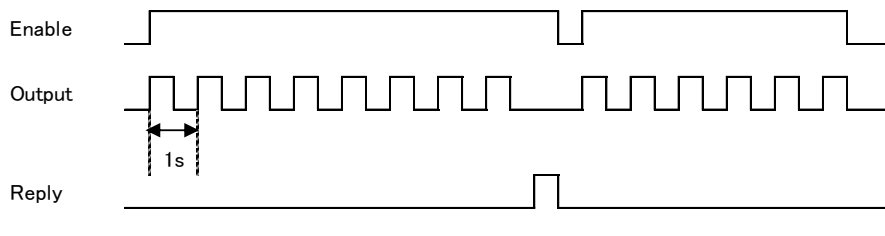
| | | |
|------|--------|---------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| BOOL | Error | : エラーフラグ (0: 正常, 1: 異常) |
| BOOL | Output | : パルス出力 |

説明

本 FB は入力 Enable を ON すると Period で指定した周期で、Count で指定したカウント数パルスを出力します。
 パルス出力完了で、Done が ON します。
 Enable が ON から OFF した時、Output は OFF されます。
 Output のパルス周期は Period に対してタスク周期 (ms) の誤差が生じます。

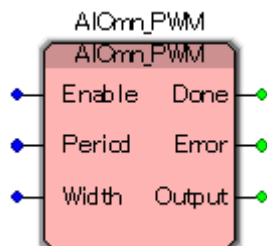
動作

・パルス出力 (Period=1s Count=8)



AIOmn_PWM 関数

機能 指定した時間幅のパルスを指定した周期で出力します

書式

入力

| | | |
|------|--------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| UINT | Period | : パルス周期 (秒) |
| UINT | Width | : パルス幅 (秒) |

出力

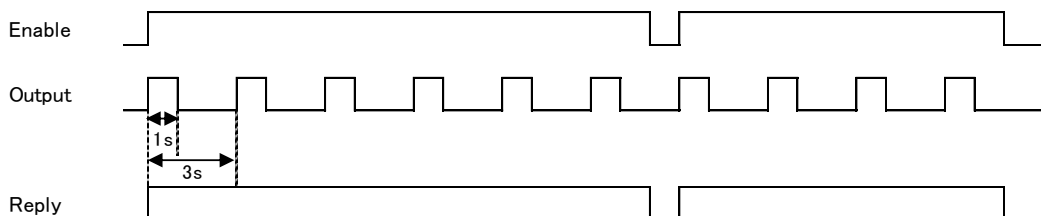
| | | |
|------|--------|---------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| BOOL | Error | : エラーフラグ (0: 正常, 1: 異常) |
| BOOL | Output | : パルス出力 |

説明

本 FB は入力 Enable を ON すると Period で指定した周期で、Width で指定したパルス幅でパルスを出力します。
 Enable が OFF から ON した時、Done が ON します。
 Enable が ON から OFF した時、Output は OFF されます。
 Output のパルス周期は Period に対してタスク周期 (ms) の誤差が生じます。

動作

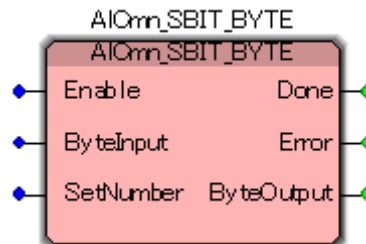
・パルス出力 (Period=3s Width=1s)



AIOmn_SBIT_BYTE 関数

機能 BYTE 型データの N ビット目をセットします

書式



入力

| | | |
|------|-----------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| BYTE | ByteInput | : 入力データ |
| UINT | SetNumber | : セットするビット番号 (0~7) |

出力

| | | |
|------|------------|---------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| BOOL | Error | : エラーフラグ (0: 正常, 1: 異常) |
| BYTE | ByteOutput | : 出力データ |

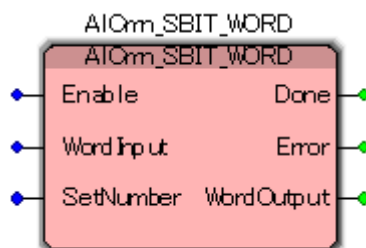
説明

本 FB は入力 Enable を ON すると WORD 型データの ByteInput の SetNumber で指定されたビットをセットし、結果を ByteOutput へ出力します。
 本 FB は入力 Enable が TRUE の間、処理を続けます。
 SetNumber は SetNumber=0 で 0 ビット目、5 で 5 ビット目を示します。

AICmn_SBIT_WORD 関数

機能 WORD 型データの N ビット目をセットします

書式



入力

| | | |
|------|-----------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| WORD | WordInput | : 入力データ |
| UINT | SetNumber | : セットするビット番号 (0~15) |

出力

| | | |
|------|------------|---------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| BOOL | Error | : エラーフラグ (0: 正常, 1: 異常) |
| WORD | WordOutput | : 出力データ |

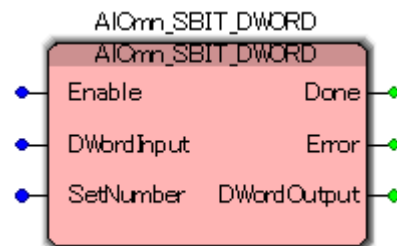
説明

本 FB は入力 Enable を ON すると WORD 型データの WordInput の SetNumber で指定されたビットをセットし、結果を WordOutput へ出力します。
本 FB は入力 Enable が TRUE の間、処理を続けます。
SetNumber は SetNumber=0 で 0 ビット目、5 で 5 ビット目を示します。

AIOmn_SBIT_DWORD 関数

機能 DWORD 型データの N ビット目をセットします

書式



入力

| | | |
|-------|------------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| DWORD | DWordInput | : 入力データ |
| UINT | SetNumber | : セットするビット番号 (0~31) |

出力

| | | |
|-------|-------------|---------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| BOOL | Error | : エラーフラグ (0: 正常, 1: 異常) |
| DWORD | DWordOutput | : 出力データ |

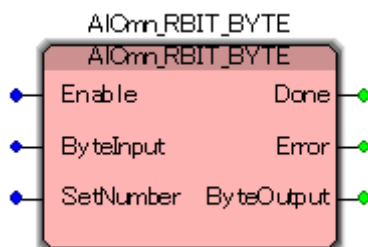
説明

本 FB は入力 Enable を ON すると DWORD 型データの DWordInput の SetNumber で指定されたビットをセットし、結果を DWordOutput へ出力します。
 本 FB は入力 Enable が TRUE の間、処理を続けます。
 SetNumber は SetNumber=0 で 0 ビット目、5 で 5 ビット目を示します。

AIOmn_RBIT_BYTE 関数

機能

BYTE 型データの N ビット目をリセットします

書式**入力**

BOOL Enable : アクションフラグ(0:停止, 1:実行)
 BYTE ByteInput : 入力データ
 UINT SetNumber : リセットするビット番号(0~7)

出力

BOOL Done : リターン(0:応答なし, 1:実行終了)
 BOOL Error : エラーフラグ(0:正常, 1:異常)
 BYTE ByteOutput : 出力データ

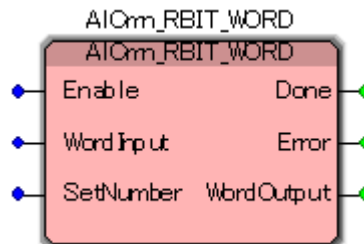
説明

本 FB は入力 Enable を ON すると BYTE 型データの ByteInput の SetNumber で指定されたビットをリセットし、結果を ByteOutput へ出力します。
 本 FB は入力 Enable が TRUE の間、処理を続けます。
 SetNumber は SetNumber=0 で 0 ビット目、5 で 5 ビット目を示します。

AIOmn_RBIT_WORD 関数

機能 WORD 型データの N ビット目をリセットします

書式



入力

| | | |
|------|-----------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| WORD | WordInput | : 入力データ |
| UINT | SetNumber | : リセットするビット番号 (0~15) |

出力

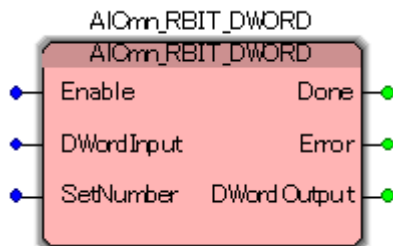
| | | |
|------|------------|---------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| BOOL | Error | : エラーフラグ (0: 正常, 1: 異常) |
| WORD | WordOutput | : 出力データ |

説明 本 FB は入力 Enable を ON すると WORD 型データの WordInput の SetNumber で指定されたビットをリセットし、結果を WordOutput へ出力します。
本 FB は入力 Enable が TRUE の間、処理を続けます。
SetNumber は SetNumber=0 で 0 ビット目、5 で 5 ビット目を示します。

AIOmn_RBIT_DWORD 関数

機能

DWORD 型データの N ビット目をリセットします

書式**入力**

BOOL Enable : アクションフラグ (0: 停止, 1: 実行)
 DWORD DWordInput : 入力データ
 UINT SetNumber : リセットするビット番号 (0~31)

出力

BOOL Done : リターン (0: 応答なし, 1: 実行終了)
 BOOL Error : エラーフラグ (0: 正常, 1: 異常)
 DWORD DWordOutput : 出力データ

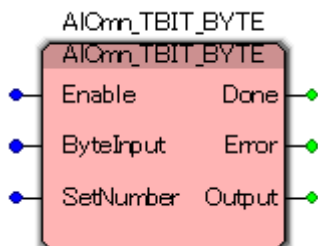
説明

本 FB は入力 Enable を ON すると DWORD 型データの DWordInput の SetNumber で指定されたビットをリセットし、結果を DWordOutput へ出力します。
 本 FB は入力 Enable が TRUE の間、処理を続けます。
 SetNumber は SetNumber=0 で 0 ビット目、5 で 5 ビット目を示します。

AIOmn_TBIT_BYTE

機能

BYTE の N ビット目をテスト(0 or 1 を判断)します

書式**入力**

BOOL Enable : アクションフラグ(0:停止, 1:実行)
 BYTE ByteInput : BYTE 型入力データ
 UINT SetNumber : テストするビット番号(0~7)

出力

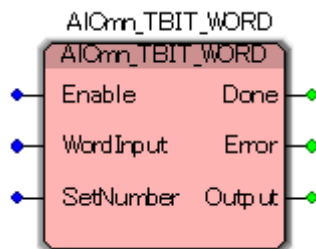
BOOL Done : リターン(0:応答なし, 1:実行終了)
 BOOL Error : エラーフラグ(0:正常, 1:異常)
 BOOL Output : テスト結果

説明

本 FB は入力 Enable を ON すると BYTE 型データの WordInput の SetNumber で指定されたビットをテスト(0 or 1 を判断)し、結果を Output へ出力します。本 FB は入力 Enable が TRUE の間、処理を続けます。SetNumber は SetNumber=0 で 0 ビット目、5 で 5 ビット目を示します。

AIOmn_TBIT_WORD 関数

機能 WORD 型データの N ビット目をテスト (0 or 1 を判断) します

書式

入力

| | | |
|------|-----------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| WORD | WordInput | : WORD 型入力データ |
| UINT | SetNumber | : テストするビット番号 (0~15) |

出力

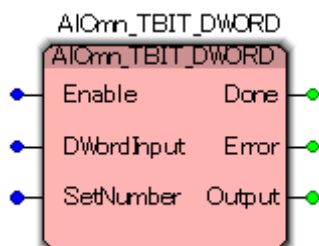
| | | |
|------|--------|---------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| BOOL | Error | : エラーフラグ (0: 正常, 1: 異常) |
| BOOL | Output | : テスト結果 |

説明 本 FB は入力 Enable を ON すると WORD 型データの WordInput の SetNumber で指定されたビットをテスト (0 or 1 を判断) し、結果を Output へ出力します。本 FB は入力 Enable が TRUE の間、処理を続けます。SetNumber は SetNumber=0 で 0 ビット目、5 で 5 ビット目を示します。

AICmn_TBIT_DWORD 関数

機能

DWORD 型データの N ビット目をテスト(0 or 1 を判断)します

書式**入力**

BOOL Enable : アクションフラグ(0:停止, 1:実行)
 DWORD DWordInput : DWORD 型入力データ
 UINT SetNumber : テストするビット番号(0~31)

出力

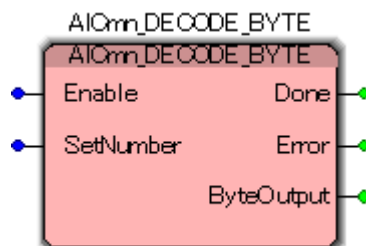
BOOL Done : リターン(0:応答なし, 1:実行終了)
 BOOL Error : エラーフラグ(0:正常, 1:異常)
 BOOL Output : テスト結果

説明

本 FB は入力 Enable を ON すると DWORD 型データの DWordInput の SetNumber で指定されたビットをテスト(0 or 1 を判断)し、結果を Output へ出力します。本 FB は入力 Enable が TRUE の間、処理を続けます。SetNumber は SetNumber=0 で 0 ビット目、5 で 5 ビット目を示します。

AICmn_DECODE_BYTE 関数

機能 指定したビットを ON した BYTE 型データを出力します

書式

入力

| | | |
|------|-----------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| UINT | SetNumber | : テストするビット番号 (0~7) |

出力

| | | |
|------|------------|---------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| DINT | Error | : エラーコード (0: 正常, 1: 異常) |
| BYTE | ByteOutput | : 出力データ |

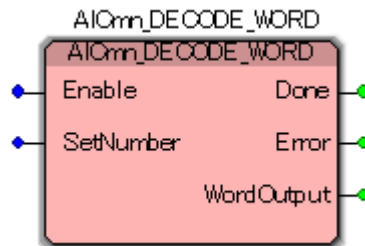
説明

本 FB は入力 Enable を ON すると SetNumber で指定されたビットのみを ON した BYTE 型データを Output へ出力します。
 本 FB は入力 Enable が TRUE の間、処理を続けます。
 SetNumber は SetNumber=0 で 0 ビット目、5 で 5 ビット目を示します。

AIOmn_DECODE_WORD 関数

機能 指定したビットを ON した WORD 型データを出力します

書式



入力

| | | |
|------|-----------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| UINT | SetNumber | : テストするビット番号 (0~15) |

出力

| | | |
|------|------------|---------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| DINT | Error | : エラーコード (0: 正常, 1: 異常) |
| WORD | WordOutput | : 出力データ |

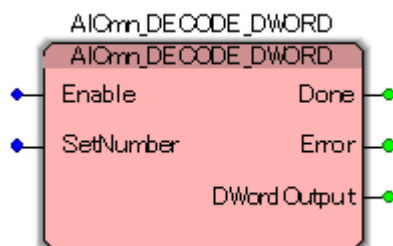
説明

本 FB は入力 Enable を ON すると SetNumber で指定されたビットのみを ON した WORD 型データを Output へ出力します。
 本 FB は入力 Enable が TRUE の間、処理を続けます。
 SetNumber は SetNumber=0 で 0 ビット目、5 で 5 ビット目を示します。

AIOmn_DECODE_DWORD 関数

機能

指定したビットを ON した DWORD 型データを出力します

書式**入力**

BOOL Enable : アクションフラグ (0: 停止, 1: 実行)
 UINT SetNumber : テストするビット番号 (0~31)

出力

BOOL Done : リターン (0: 応答なし, 1: 実行終了)
 DINT Error : エラーコード (0: 正常, 1: 異常)
 DWORD DWordOutput : 出力データ

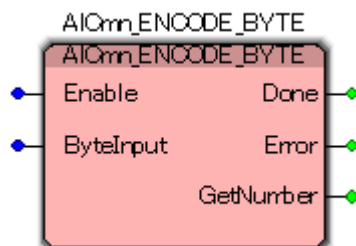
説明

本 FB は入力 Enable を ON すると SetNumber で指定されたビットのみを ON した DWORD 型データを Output へ出力します。
 本 FB は入力 Enable が TRUE の間、処理を続けます。
 SetNumber は SetNumber=0 で 0 ビット目、5 で 5 ビット目を示します。

AIOmn_ENCODE_BYTE 関数

機能

BYTE 型データの内、最も下位の ON しているビット番号を出力します

書式**入力**

BOOL Enable : アクションフラグ (0: 停止, 1: 実行)
 BYTE ByteInput : 入力データ

出力

BOOL Done : リターン (0: 応答なし, 1: 実行終了)
 BOOL Error : エラーフラグ (0: 正常, 1: 異常)
 UINT GetNumber : 最も下位側の ON しているビット番号

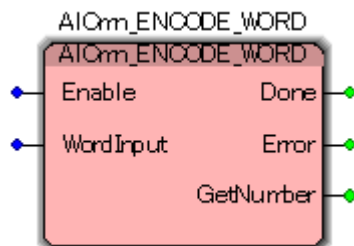
説明

本 FB は入力 Enable を ON すると BYTE 型データの Input の内で最も下位側の ON しているビットの番号を GetNumber へ出力します。
 本 FB は入力 Enable が TRUE の間、処理を続けます。
 GetNumber は GetNumber=0 で 0 ビット目、GetNumber=5 で 5 ビット目を示します。

AICmn_ENCODE_WORD 関数

機能

WORD 型データの内、最も下位の ON しているビット番号を出力します

書式**入力**

BOOL Enable : アクションフラグ (0:停止, 1:実行)
WORD WordInput : 入力データ

出力

BOOL Done : リターン (0:応答なし, 1:実行終了)
BOOL Error : エラーフラグ (0:正常, 1:異常)
UINT GetNumber : 最も下位側の ON しているビット番号

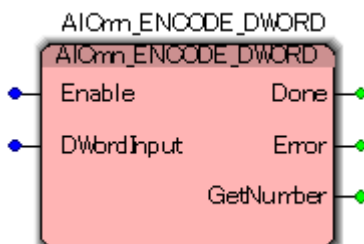
説明

本 FB は入力 Enable を ON すると WORD 型データの Input の内で最も下位側の ON しているビットの番号を GetNumber へ出力します。
本 FB は入力 Enable が TRUE の間、処理を続けます。
GetNumber は GetNumber=0 で 0 ビット目、GetNumber=5 で 5 ビット目を示します。

AICmn_ENCODE_DWORD 関数

機能

DWORD 型データの内、最も下位の ON しているビット番号を出力します

書式**入力**

BOOL Enable : アクションフラグ (0: 停止, 1: 実行)
 DWORD DWordInput : 入力データ

出力

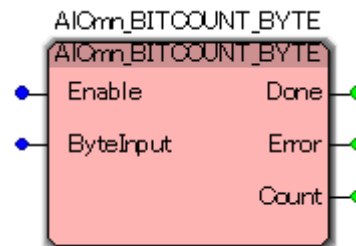
BOOL Done : リターン (0: 応答なし, 1: 実行終了)
 BOOL Error : エラーフラグ (0: 正常, 1: 異常)
 UINT GetNumber : 最も下位側の ON しているビット番号

説明

本 FB は入力 Enable を ON すると DWORD 型データの Input の内で最も下位側の ON しているビットの番号を GetNumber へ出力します。
 本 FB は入力 Enable が TRUE の間、処理を続けます。
 GetNumber は GetNumber=0 で 0 ビット目、GetNumber=5 で 5 ビット目を示します。

AIOmn_BITCOUNT_BYTE 関数

機能 BYTE 型データの内、ON しているビットの個数を出力します

書式

入力

| | | |
|------|-----------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| BYTE | ByteInput | : 入力データ |

出力

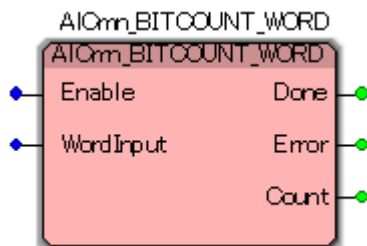
| | | |
|------|-------|---------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| BOOL | Error | : エラーフラグ (0: 正常, 1: 異常) |
| UINT | Count | : ON しているビットの個数 |

説明

本 FB は入力 Enable を ON すると BYTE 型データの Input の内で ON しているビットの個数を Count へ出力します。
 本 FB は入力 Enable が TRUE の間、処理を続けます。

AICmn_BITCOUNT_WORD 関数

機能 WORD 型データの内、ON しているビットの個数を出力します

書式

入力

| | | |
|------|-----------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| WORD | WordInput | : 入力データ |

出力

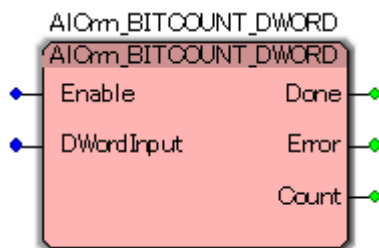
| | | |
|------|-------|---------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| BOOL | Error | : エラーフラグ (0: 正常, 1: 異常) |
| UINT | Count | : ON しているビットの個数 |

説明 本 FB は入力 Enable を ON すると WORD 型データの Input の内で ON しているビットの個数を Count へ出力します。
本 FB は入力 Enable が TRUE の間、処理を続けます。

AIOmn_BITCOUNT_DWORD 関数

機能

DWORD 型データの内、ON しているビットの個数を出力します

書式**入力**

BOOL Enable : アクションフラグ (0: 停止, 1: 実行)
 DWORD DWordInput : 入力データ

出力

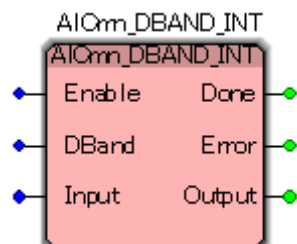
BOOL Done : リターン (0: 応答なし, 1: 実行終了)
 BOOL Error : エラーフラグ (0: 正常, 1: 異常)
 UINT Count : ON しているビットの個数

説明

本 FB は入力 Enable を ON すると DWORD 型データの Input の内で ON しているビットの個数を Count へ出力します。
 本 FB は入力 Enable が TRUE の間、処理を続けます。

AIOmn_DBAND_INT 関数

機能 指定した INT 型データを不感帯幅として、出力が変化します

書式

入力

| | | |
|------|--------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| INT | DBand | : 不感帯幅 |
| INT | Input | : 入力データ |

出力

| | | |
|------|--------|---------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| BOOL | Error | : エラーフラグ (0: 正常, 1: 異常) |
| INT | Output | : 出力データ |

説明 本 FB は入力 Enable を ON すると DBand を不感帯幅とし、以下の条件で出力します。

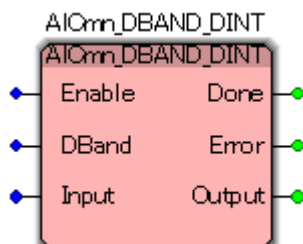
- ① $Input > DBand$ の場合、 $Input - DBand$ を出力
- ② $Input < -DBand$ の場合、 $Input + DBand$ を出力
- ③ 上記以外の場合、0 を出力

本 FB は入力 Enable が TRUE の間、処理を続けます。

AICmn_DBAND_DINT 関数

機能

指定した DINT 型データを不感帯幅として、出力が変化します

書式**入力**

| | | |
|------|--------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| DINT | DBand | : 不感帯幅 |
| DINT | Input | : 入力データ |

出力

| | | |
|------|--------|---------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| BOOL | Error | : エラーフラグ (0: 正常, 1: 異常) |
| DINT | Output | : 出力データ |

説明

本 FB は入力 Enable を ON すると DBand を不感帯幅とし、以下の条件で出力します。

- ① $Input > DBand$ の場合、 $Input - DBand$ を出力
- ② $Input < -DBand$ の場合、 $Input + DBand$ を出力
- ③ 上記以外の場合、0 を出力

本 FB は入力 Enable が TRUE の間、処理を続けます。

AIQmn_DBAND_REAL 関数

機能

指定した REAL 型データを不感帯幅として、出力が変化します

書式**入力**

| | | |
|------|--------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| REAL | DBand | : 不感帯幅 |
| REAL | Input | : 入力データ |

出力

| | | |
|------|--------|---------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| BOOL | Error | : エラーフラグ (0: 正常, 1: 異常) |
| REAL | Output | : 出力データ |

説明

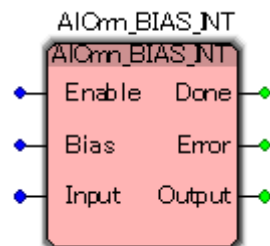
本 FB は入力 Enable を ON すると DBand を不感帯幅とし、以下の条件で出力します。

- ① $Input > DBand$ の場合、 $Input - DBand$ を出力
- ② $Input < -DBand$ の場合、 $Input + DBand$ を出力
- ③ 上記以外の場合、0 を出力

本 FB は入力 Enable が TRUE の間、処理を続けます。

AIOmn_BIAS_INT 関数

機能 指定した INT 型データをバイアスとして、出力が変化します

書式

入力

| | | |
|------|--------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| INT | Bias | : バイアス |
| INT | Input | : 入力データ |

出力

| | | |
|------|--------|---------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| BOOL | Error | : エラーフラグ (0: 正常, 1: 異常) |
| INT | Output | : 出力データ |

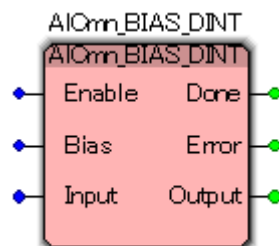
説明 本 FB は入力 Enable を ON すると Bias をバイアスとし、以下の条件で出力します。

- ① Input > 0 の場合、Input + Bias を出力
- ② Input < 0 の場合、Input - Bias を出力
- ③ Input = 0 の場合、0 を出力

バイアスした結果が INT 型の上限値超過の場合は、上限値 (32, 767) が出力されます。
 バイアスした結果が INT 型の下限値未満の場合は、下限値 (-32, 768) が出力されます。
 本 FB は入力 Enable が TRUE の間、処理を続けます。

AIOmn_BIAS_DINT 関数

機能 指定した DINT 型データをバイアスとして、出力が変化します

書式

入力

| | | |
|------|--------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| DINT | Bias | : バイアス |
| DINT | Input | : 入力データ |

出力

| | | |
|------|--------|---------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| BOOL | Error | : エラーフラグ (0: 正常, 1: 異常) |
| DINT | Output | : 出力データ |

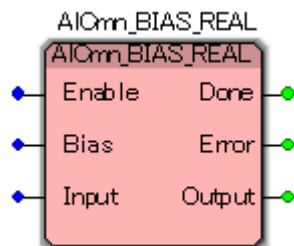
説明 本 FB は入力 Enable を ON すると Bias をバイアスとし、以下の条件で出力します。

- ① Input > 0 の場合、Input + Bias を出力
- ② Input < 0 の場合、Input - Bias を出力
- ③ Input = 0 の場合、0 を出力

バイアスした結果が INT 型の上限値超過の場合は、上限値 (2, 147, 483, 647) が出力されます。
 バイアスした結果が INT 型の下限値未満の場合は、下限値 (-2, 147, 483, 648) が出力されます。
 本 FB は入力 Enable が TRUE の間、処理を続けます。

AIOmn_BIAS_REAL 関数

機能 指定した REAL 型データをバイアスとして、出力が変化します

書式

入力

| | | |
|------|--------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| REAL | Bias | : バイアス |
| REAL | Input | : 入力データ |

出力

| | | |
|------|--------|---------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| BOOL | Error | : エラーフラグ (0: 正常, 1: 異常) |
| REAL | Output | : 出力データ |

説明 本 FB は入力 Enable を ON すると Bias をバイアスとし、以下の条件で出力します。

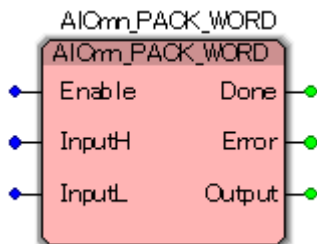
- ① Input > 0 の場合、Input + Bias を出力
- ② Input < 0 の場合、Input - Bias を出力
- ③ Input = 0 の場合、0 を出力

本 FB は入力 Enable が TRUE の間、処理を続けます。

AIOmn_PACK_WORD 関数

機能

2つの BYTE 型データを1つの WORD 型データに結合し、出力します

書式**入力**

BOOL Enable : アクションフラグ(0:停止, 1:実行)
 BYTE InputH : 入力データ HI
 BYTE InputL : 入力データ LO

出力

BOOL Done : リターン(0:応答なし, 1:実行終了)
 BOOL Error : エラーフラグ(0:正常, 1:異常)
 WORD Output : 出力データ

説明

本FBは入力 Enable を ON すると InputH と InputL に入力された2つの BYTE 型データを1つの WORD 型データに結合します。

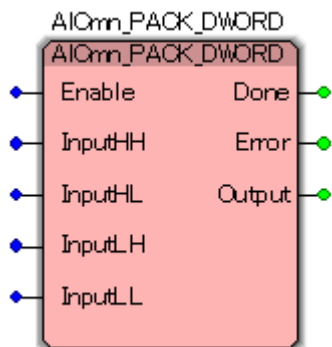
InputH で指定したデータをビット 15-8 に InputL で指定したデータをビット 7-0 に格納し、Output に出力します。

本FBは入力 Enable が TRUE の間、処理を続けます。

AIOmn_PACK_DWORD 関数

機能

4つの BYTE 型データを1つの DWORD 型データに結合し、出力します

書式**入力**

| | | |
|------|---------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| BYTE | InputHH | : 入力データ HH |
| BYTE | InputHL | : 入力データ HL |
| BYTE | InputLH | : 入力データ LH |
| BYTE | InputLL | : 入力データ LL |

出力

| | | |
|-------|--------|---------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| BOOL | Error | : エラーフラグ (0: 正常, 1: 異常) |
| DWORD | Output | : 出力データ |

説明

本 FB は入力 Enable を ON すると InputHH、InputHL、InputLH、InputLL に入力された 4 つの BYTE 型データを 1 つの DWORD 型データに結合します。

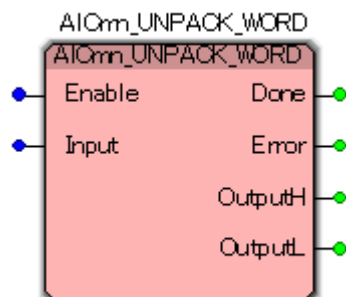
InputHH で指定したデータをビット 31-24 に InputHL で指定したデータをビット 23-16 に InputLH で指定したデータをビット 15-8 に InputLL で指定したデータをビット 7-0 に格納し、Output に出力します。

本 FB は入力 Enable が TRUE の間、処理を続けます。

AICmn_UNPACK_WORD 関数

機能

1つの WORD 型データを2つの BYTE 型データに分離し、出力します

書式**入力**

BOOL Enable : アクションフラグ(0:停止, 1:実行)
WORD Data : 入力データ

出力

BOOL Done : リターン(0:応答なし, 1:実行終了)
BOOL Error : エラーフラグ(0:正常, 1:異常)
BYTE OutputH : 出力データ H
BYTE OutputL : 出力データ L

説明

本 FB は入力 Enable を ON すると Input に入力された1つの WORD 型データを2つの BYTE 型データに分離します。

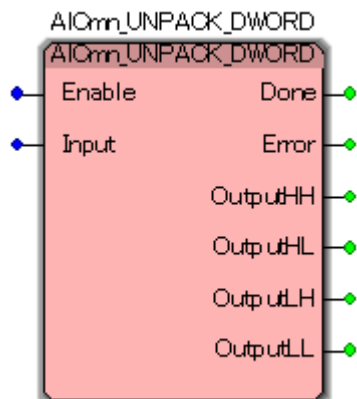
Input で指定したデータのビット 15-8 を OutputH に、Input で指定したデータのビット 7-0 を OutputL に格納し、出力します。

本 FB は入力 Enable が TRUE の間、処理を続けます。

AICmn_UNPACK_DWORD 関数

機能

1つのDWORD型データを4つのBYTE型データに分離し、出力します

書式**入力**

| | | |
|-------|--------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| DWORD | Data | : 書込データ |
| DWORD | Input | : 入力データ |

出力

| | | |
|------|----------|---------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| BOOL | Error | : エラーフラグ (0: 正常, 1: 異常) |
| BYTE | OutputHH | : 出力データ HH |
| BYTE | OutputHL | : 出力データ HL |
| BYTE | OutputLH | : 出力データ LH |
| BYTE | OutputLL | : 出力データ LL |

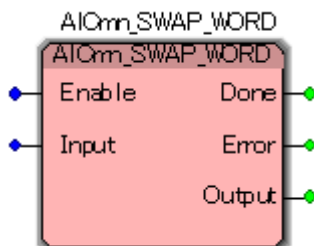
説明

本FBは入力EnableをONするとInputに入力された1つのDWORD型データを4つのBYTE型データに分離します。
 Inputで指定したデータのビット31-24をOutputHHに、Inputで指定したデータのビット23-16をOutputHLに、Inputで指定したデータのビット15-8をOutputLHに、Inputで指定したデータのビット7-0をOutputLLに分解し、出力します。
 本FBは入力EnableがTRUEの間、処理を続けます。

AIOmn_SWAP_WORD 関数

機能

WORD 型データの上位バイトと下位バイトを入れ替え出力します

書式**入力**

BOOL Enable : アクションフラグ (0: 停止, 1: 実行)
WORD Input : 入力データ

出力

BOOL Done : リターン (0: 応答なし, 1: 実行終了)
BOOL Error : エラーフラグ (0: 正常, 1: 異常)
WORD Output : 出力データ

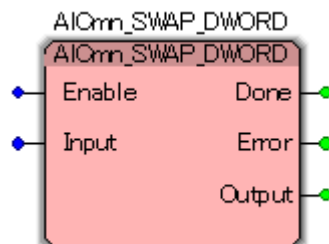
説明

本 FB は入力 Enable を ON すると Input に入力された WORD 型データのビット 15-8 とビット 7-0 を入れ替えて、Output に出力します。
本 FB は入力 Enable が TRUE の間、処理を続けます。

AIOmn_SWAP_DWORD 関数

機能

DWORD 型データの上位バイトと下位バイトを入れ替え出力します

書式**入力**

BOOL Enable : アクションフラグ (0: 停止, 1: 実行)
 DWORD Input : 入力データ

出力

BOOL Done : リターン (0: 応答なし, 1: 実行終了)
 BOOL Error : エラーフラグ (0: 正常, 1: 異常)
 DWORD Output : 出力データ

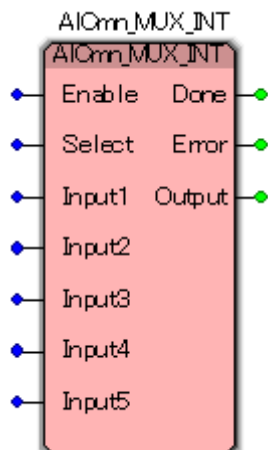
説明

本 FB は入力 Enable を ON すると Input に入力された DWORD 型データのビット 31-24 をビット 7-0 へ、ビット 23-16 をビット 15-8 へ、ビット 15-8 をビット 23-16 へ、ビット 7-0 をビット 31-24 への入れ替えを行い、Output に出力します。
 本 FB は入力 Enable が TRUE の間、処理を続けます。

AIOmn_MUX_INT 関数

機能

複数の INT 型データの中から選択した 1 つの INT 型データを出力します

書式**入力**

| | | |
|------|--------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| INT | Select | : セレクター (0~4) |
| INT | Input1 | : 入力データ 1 |
| INT | Input2 | : 入力データ 2 |
| INT | Input3 | : 入力データ 3 |
| INT | Input4 | : 入力データ 4 |
| INT | Input5 | : 入力データ 5 |

出力

| | | |
|------|--------|---------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| BOOL | Error | : エラーフラグ (0: 正常, 1: 異常) |
| INT | Output | : 出力データ |

説明

本 FB は入力 Enable を ON すると Select に入力された値に従って、対応する入力データを Output に出力します。

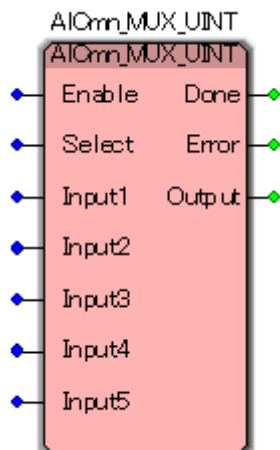
Select=0 は Input1 に対応し、Select=4 は Input5 に対応します。

本 FB は入力 Enable が TRUE の間、処理を続けます。

AICmn_MUX_UINT 関数

機能

複数の UINT 型データの中から選択した 1 つの UINT 型データを出力します

書式**入力**

| | | |
|------|--------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| INT | Select | : セレクター (0~4) |
| UINT | Input1 | : 入力データ 1 |
| UINT | Input2 | : 入力データ 2 |
| UINT | Input3 | : 入力データ 3 |
| UINT | Input4 | : 入力データ 4 |
| UINT | Input5 | : 入力データ 5 |

出力

| | | |
|------|--------|---------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| BOOL | Error | : エラーフラグ (0: 正常, 1: 異常) |
| UINT | Output | : 出力データ |

説明

本 FB は入力 Enable を ON すると Select に入力された値に従って、対応する入力データを Output に出力します。

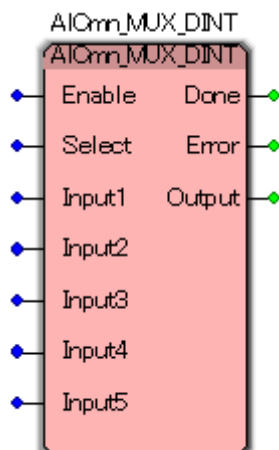
Select=0 は Input1 に対応し、Select=4 は Input5 に対応します。

本 FB は入力 Enable が TRUE の間、処理を続けます。

AIOmn_MUX_DINT 関数

機能

複数の DINT 型データの中から選択した 1 つの DINT 型データを出力します

書式**入力**

| | | |
|------|--------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| INT | Select | : セレクター (0~4) |
| DINT | Input1 | : 入力データ 1 |
| DINT | Input2 | : 入力データ 2 |
| DINT | Input3 | : 入力データ 3 |
| DINT | Input4 | : 入力データ 4 |
| DINT | Input5 | : 入力データ 5 |

出力

| | | |
|------|--------|---------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| BOOL | Error | : エラーフラグ (0: 正常, 1: 異常) |
| DINT | Output | : 出力データ |

説明

本 FB は入力 Enable を ON すると Select に入力された値に従って、対応する入力データを Output に出力します。

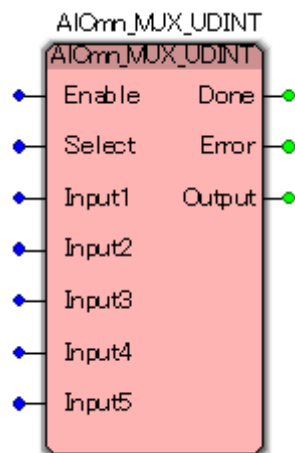
Select=0 は Input1 に対応し、Select=4 は Input5 に対応します。

本 FB は入力 Enable が TRUE の間、処理を続けます。

AIOmn_MUX_UDINT 関数

機能

複数の UDINT 型データの中から選択した 1 つの UDINT 型データを出力します

書式**入力**

| | | |
|-------|--------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| INT | Select | : セレクター (0~4) |
| UDINT | Input1 | : 入力データ 1 |
| UDINT | Input2 | : 入力データ 2 |
| UDINT | Input3 | : 入力データ 3 |
| UDINT | Input4 | : 入力データ 4 |
| UDINT | Input5 | : 入力データ 5 |

出力

| | | |
|-------|--------|---------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| BOOL | Error | : エラーフラグ (0: 正常, 1: 異常) |
| UDINT | Output | : 出力データ |

説明

本 FB は入力 Enable を ON すると Select に入力された値に従って、対応する入力データを Output に出力します。

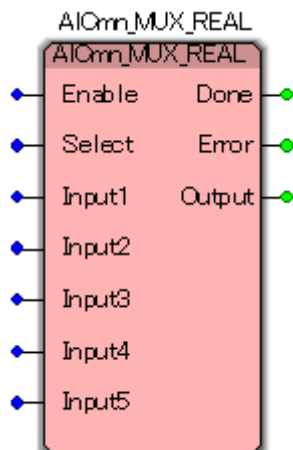
Select=0 は Input1 に対応し、Select=4 は Input5 に対応します。

本 FB は入力 Enable が TRUE の間、処理を続けます。

AIOmn_MUX_REAL 関数

機能

複数の REAL 型データの中から選択した 1 つの REAL 型データを出力します

書式**入力**

| | | |
|------|--------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| INT | Select | : セレクター (0~4) |
| REAL | Input1 | : 入力データ 1 |
| REAL | Input2 | : 入力データ 2 |
| REAL | Input3 | : 入力データ 3 |
| REAL | Input4 | : 入力データ 4 |
| REAL | Input5 | : 入力データ 5 |

出力

| | | |
|------|--------|---------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| BOOL | Error | : エラーフラグ (0: 正常, 1: 異常) |
| REAL | Output | : 出力データ |

説明

本 FB は入力 Enable を ON すると Select に入力された値に従って、対応する入力データを Output に出力します。

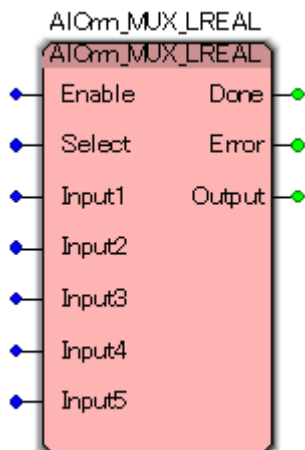
Select=0 は Input1 に対応し、Select=4 は Input5 に対応します。

本 FB は入力 Enable が TRUE の間、処理を続けます。

AIOmn_MUX_LREAL 関数

機能

複数の LREAL 型データの中から選択した 1 つの LREAL 型データを出力します

書式**入力**

| | | |
|-------|--------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| INT | Select | : セレクター (0~4) |
| LREAL | Input1 | : 入力データ 1 |
| LREAL | Input2 | : 入力データ 2 |
| LREAL | Input3 | : 入力データ 3 |
| LREAL | Input4 | : 入力データ 4 |
| LREAL | Input5 | : 入力データ 5 |

出力

| | | |
|-------|--------|---------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| BOOL | Error | : エラーフラグ (0: 正常, 1: 異常) |
| LREAL | Output | : 出力データ |

説明

本 FB は入力 Enable を ON すると Select に入力された値に従って、対応する入力データを Output に出力します。

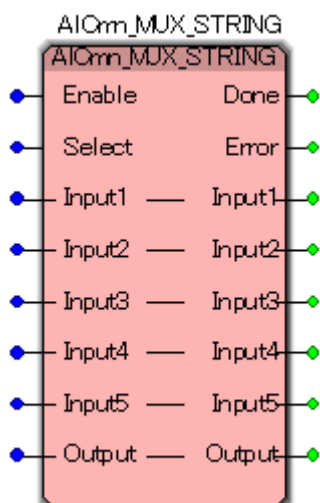
Select=0 は Input1 に対応し、Select=4 は Input5 に対応します。

本 FB は入力 Enable が TRUE の間、処理を続けます。

AIOmn_MUX_STRING 関数

機能

複数の STRING 型データの中から選択した 1 つの STRING 型データを出力します

書式**入力**

BOOL Enable : アクションフラグ (0: 停止, 1: 実行)
 INT Select : セレクター (0~4)

出力

BOOL Done : リターン (0: 応答なし, 1: 実行終了)
 BOOL Error : エラーフラグ (0: 正常, 1: 異常)

入出力

STRING Input1 : 入力データ 1
 STRING Input2 : 入力データ 2
 STRING Input3 : 入力データ 3
 STRING Input4 : 入力データ 4
 STRING Input5 : 入力データ 5
 STRING Output : 出力データ

説明

本 FB は入力 Enable を ON すると Select に入力された値に従って、対応する入力データを Output に出力します。

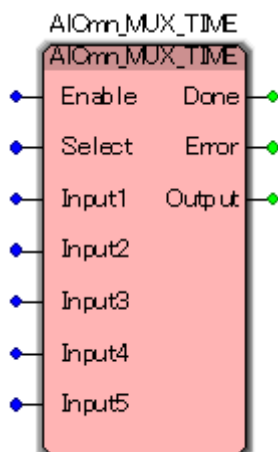
Select=0 は Input1 に対応し、Select=4 は Input5 に対応します。

本 FB は入力 Enable が TRUE の間、処理を続けます。

AIOmn_MUX_TIME 関数

機能

複数の TIME 型データから 1 つの TIME 型データを出力します

書式**入力**

| | | |
|------|--------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| INT | Select | : セレクター (0~4) |
| TIME | Input1 | : 入力データ 1 |
| TIME | Input2 | : 入力データ 2 |
| TIME | Input3 | : 入力データ 3 |
| TIME | Input4 | : 入力データ 4 |
| TIME | Input5 | : 入力データ 5 |

出力

| | | |
|------|--------|---------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| BOOL | Error | : エラーフラグ (0: 正常, 1: 異常) |
| TIME | Output | : 出力データ |

説明

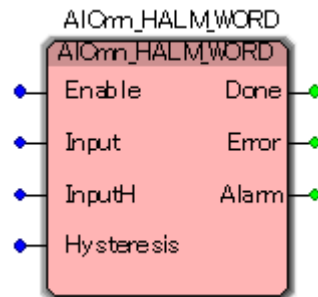
本 FB は入力 Enable を ON すると Select に入力された値に従って、対応する入力データを Output に出力します。

Select=0 は Input1 に対応し、Select=4 は Input5 に対応します。

本 FB は入力 Enable が TRUE の間、処理を続けます。

AICmn_HALM_WORD 関数

機能 入力した WORD 型データが上限設定値を超える値の場合に警報を出力します

書式

入力

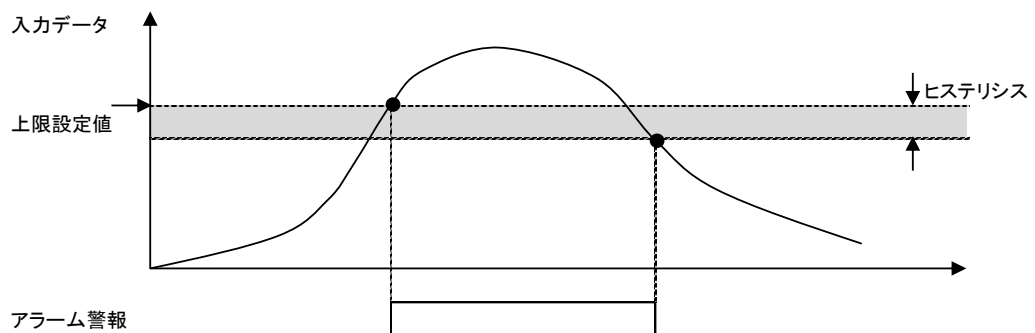
| | | |
|------|------------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| WORD | Input | : 入力データ |
| WORD | InputH | : 上限設定データ |
| WORD | Hysteresis | : ヒステリシス |

出力

| | | |
|------|-------|-----------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| BOOL | Error | : エラーフラグ (0: 正常, 1: 異常) |
| BOOL | Alarm | : アラーム警報 (0: 警報なし, 1: 警報あり) |

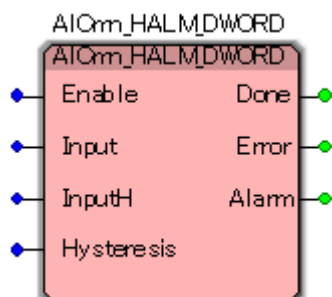
説明

本FBは入力 Enable を ON すると Input が InputH 以下であるかを監視します。
 Input が InputH を超過した場合、警報を出力します。
 Input > InputH の場合、Alarm は TRUE を出力します。
 Input < InputH - Hysteresis の場合、Alarm は FALSE を出力します。
 本FBは入力 Enable が TRUE の間、監視を続けます。



AIOmn_HALM_DWORD 関数

機能 入力した DWORD 型データが上限設定値を超える値の場合に警報を出力します

書式

入力

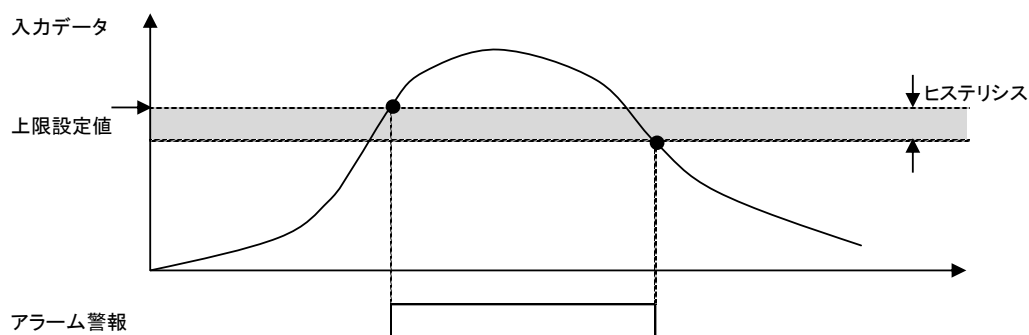
| | | |
|-------|------------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| DWORD | Input | : 入力データ |
| DWORD | InputH | : 上限設定データ |
| DWORD | Hysteresis | : ヒステリシス |

出力

| | | |
|------|-------|-----------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| BOOL | Error | : エラーフラグ (0: 正常, 1: 異常) |
| BOOL | Alarm | : アラーム警報 (0: 警報なし, 1: 警報あり) |

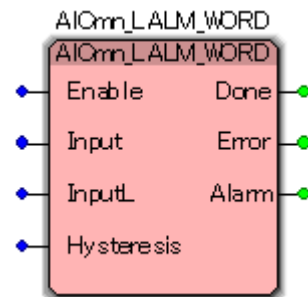
説明

本FBは入力 Enable を ON すると Input が InputH 以下であるかを監視します。
 Input が InputH を超過した場合、警報を出力します。
 Input > InputH の場合、Alarm は TRUE を出力します。
 Input < InputH - Hysteresis の場合、Alarm は FALSE を出力します。
 本FBは入力 Enable が TRUE の間、監視を続けます。



AIOmn_LALM_WORD 関数

機能 入力した WORD 型データが下限設定値未満の値の場合に警報を出力します

書式

入力

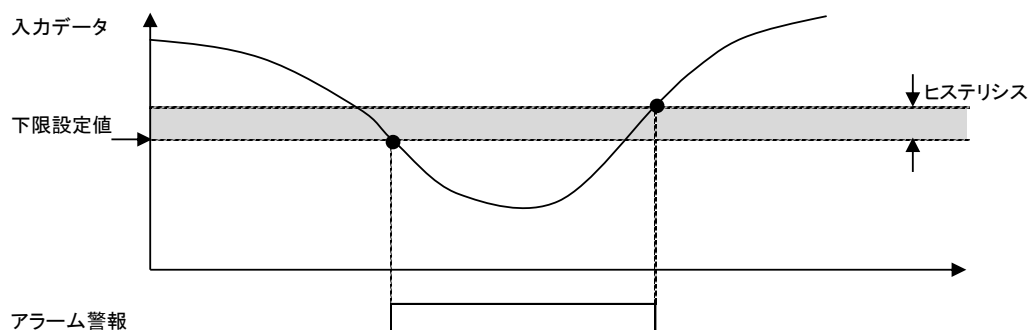
| | | |
|------|------------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| WORD | Input | : 入力データ |
| WORD | InputL | : 下限設定データ |
| WORD | Hysteresis | : ヒステリシス |

出力

| | | |
|------|-------|-----------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| BOOL | Error | : エラーフラグ (0: 正常, 1: 異常) |
| BOOL | Alarm | : アラーム警報 (0: 警報なし, 1: 警報あり) |

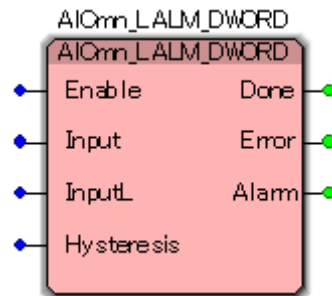
説明

本 FB は入力 Enable を ON すると Input が InputL 以上であるかを監視します。
 Input が InputL 未満の場合、警報を出力します。
 Input < InputL の場合、Alarm は TRUE を出力します。
 Input > InputL + Hysteresis の場合、Alarm は FALSE を出力します。
 本 FB は入力 Enable が TRUE の間、監視を続けます。



AIOmn_LALM_DWORD 関数

機能 入力した DWORD 型データが下限設定値未満の値の場合に警報を出力します

書式

入力

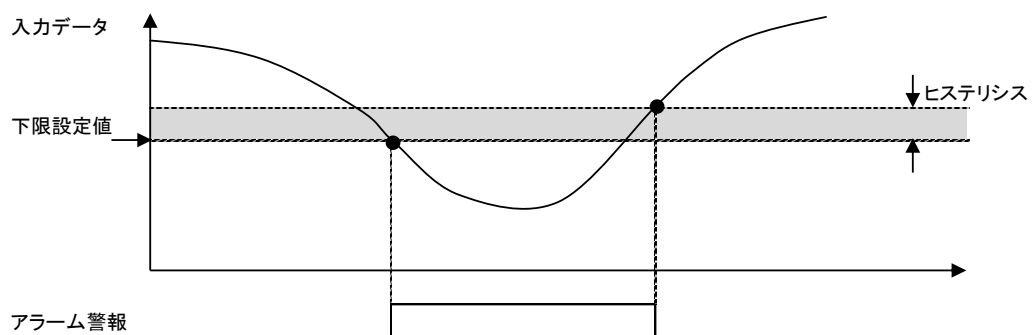
| | | |
|-------|------------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| DWORD | Input | : 入力データ |
| DWORD | InputL | : 下限設定データ |
| DWORD | Hysteresis | : ヒステリシス |

出力

| | | |
|------|-------|-----------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| BOOL | Error | : エラーフラグ (0: 正常, 1: 異常) |
| BOOL | Alarm | : アラーム警報 (0: 警報なし, 1: 警報あり) |

説明

本 FB は入力 Enable を ON すると Input が InputL 以上であるかを監視します。
 Input が InputL 未満の場合、警報を出力します。
 Input < InputL の場合、Alarm は TRUE を出力します。
 Input > InputL + Hysteresis の場合、Alarm は FALSE を出力します。
 本 FB は入力 Enable が TRUE の間、監視を続けます。

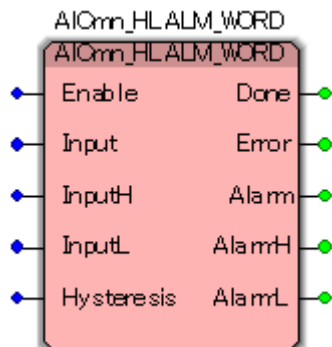


AICmn_HLALM_WORD 関数

機能

入力した WORD 型データが下限設定値未満、または上限設定値を超える値の場合に警報を出力します

書式



入力

| | | |
|------|------------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| WORD | Input | : 入力データ |
| WORD | InputH | : 上限設定データ |
| WORD | InputL | : 下限設定データ |
| WORD | Hysteresis | : ヒステリシス |

出力

| | | |
|------|-------|-----------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| BOOL | Error | : エラーフラグ (0: 正常, 1: 異常) |
| BOOL | Alarm | : アラーム警報 (0: 警報なし, 1: 警報あり) |

説明

本 FB は入力 Enable を ON すると Input が InputH 以下、または InputL 以上であるかを監視します。

Input が InputH を超過した場合、または InputL 未満の場合、警報を出力します。

Input > InputH または Input < InputL の場合、Alarm は TRUE を出力します。

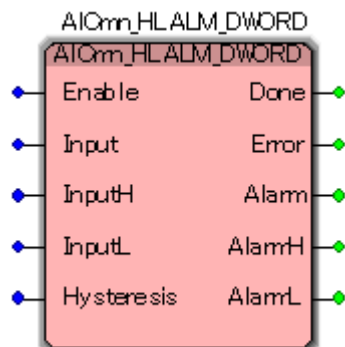
Input < InputH - Hysteresis または Input > InputL + Hysteresis の場合、Alarm は FALSE を出力します。

本 FB は入力 Enable が TRUE の間、監視を続けます。

AIOmn_HLALM_DWORD 関数

機能

入力した DWORD 型データが下限設定値未満、または上限設定値を超える値の場合に警報を出力します

書式**入力**

| | | |
|-------|------------|---------------------------|
| BOOL | Enable | : アクションフラグ (0: 停止, 1: 実行) |
| DWORD | Input | : 入力データ |
| DWORD | InputH | : 上限設定データ |
| DWORD | InputL | : 下限設定データ |
| DWORD | Hysteresis | : ヒステリシス |

出力

| | | |
|------|-------|-----------------------------|
| BOOL | Done | : リターン (0: 応答なし, 1: 実行終了) |
| BOOL | Error | : エラーフラグ (0: 正常, 1: 異常) |
| BOOL | Alarm | : アラーム警報 (0: 警報なし, 1: 警報あり) |

説明

本 FB は入力 Enable を ON すると Input が InputH 以下、または InputL 以上であるかを監視します。

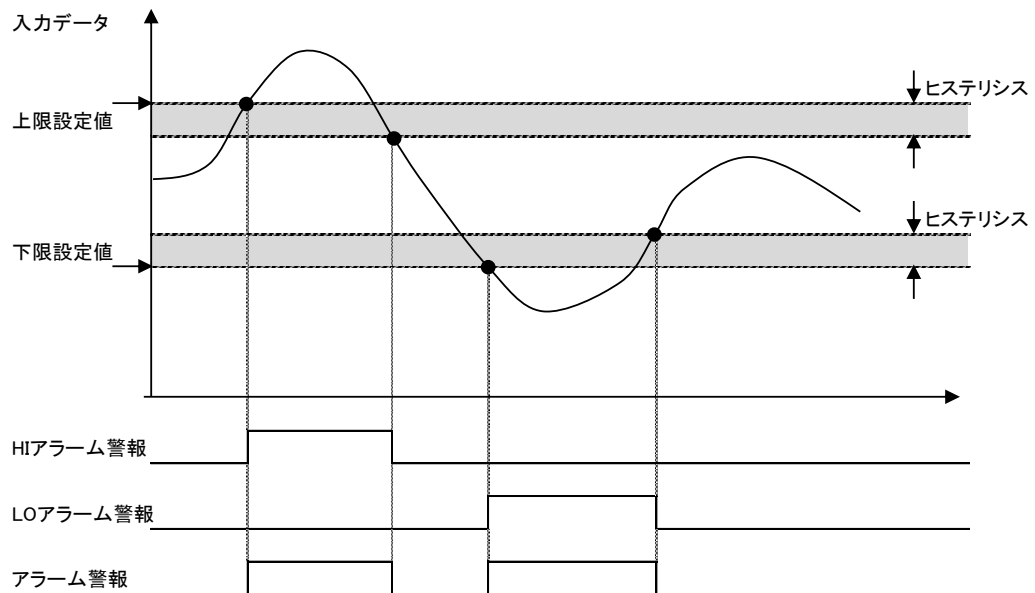
Input が InputH を超過した場合、または InputL 未満の場合、警報を出力します。

Input > InputH または Input < InputL の場合、Alarm は TRUE を出力します。

Input < InputH - Hysteresis または Input > InputL + Hysteresis の場合、Alarm は FALSE を出力します。

本 FB は入力 Enable が TRUE の間、監視を続けます。

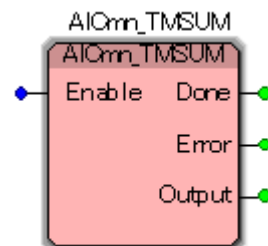
● AICmn_HLALM_WORD/ AICmn_HLALM_DWORD の動作例



AIOmn_TMSUM 関数

機能 システム起動時からの経過時間を出力します

書式



入力 BOOL Enable : アクションフラグ (0: 停止, 1: 実行)

出力 BOOL Done : リターン (0: 応答なし, 1: 実行終了)

 BOOL Error : エラーフラグ (0: 正常, 1: 異常)

 TIME Output : システム起動からの経過時間 (ms)

説明 本 FB は入力 Enable を ON するとシステム起動からの経過時間を Output に出力します。

 本 FB は入力 Enable が TRUE の間、処理を続けます。

第 3 章 付録

3-1 参考文献

- 「IEC61131-3 を用いた PLC プログラミング」

| | |
|-----|----------------------------|
| 著者 | K.-H. John / M. Tiegelkamp |
| 監訳者 | PLCopen Japan |
| 発行者 | 深田 良治 |
| 発行所 | シュプリンガー・フェアラーク東京株式会社 |
| 発行年 | 2006 年 |

本 CD-ROM には PHOENIX CONTACT 社提供の MULTIPROG に関するマニュアルも収録しております。
MULTIPROG の使用方法に関する詳細などはそちらを参照してください。

このユーザーズマニュアルについて

- (1) 本書の内容の一部又は全部を当社からの事前の承諾を得ることなく、無断で複写、複製、掲載することは固くお断りします。
- (2) 本書の内容に関しては、製品改良のためお断りなく、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。
- (3) 本書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ございましたらお手数ですが巻末記載の弊社までご連絡ください。その際、巻末記載の書籍番号も併せてお知らせください。