

リファレンスマニュアル

『MULTIPROG 用 Modbus ライブラリ』

目次

はじめに

1) ……お願いと注意 ……	1
----------------	---

第1章 MULTIPROG 用 Modbus ライブラリ

1-1 MULTIPROG とは ……	1-1
1-2 Modbus とは ……	1-1
1-3 基本設定 ……	1-2
1-3-1 ハードウェア設定 ……	1-2
1-3-2 ソフトウェア設定 ……	1-2

第2章 ファンクションブロック

2-1 機能概要 ……	2-3
2-2 使用方法 ……	2-5
2-3 ファンクションブロックリファレンス ……	2-9
MBF_CommOpen FB ……	2-11
MBF_CommClose FB ……	2-13
MBF_ReadCoilStatus FB ……	2-14
MBF_ReadCoilStatusAns FB ……	2-15
MBF_ReadInputStatus FB ……	2-16
MBF_ReadInputStatusAns FB ……	2-17
MBF_ReadHoldingRgs FB ……	2-18
MBF_ReadHoldingRgsAns FB ……	2-19
MBF_ReadInputRgs FB ……	2-20
MBF_ReadInputRgsAns FB ……	2-21
MBF_ForceSingleCoil FB ……	2-22
MBF_ForceSingleCoilAns FB ……	2-23
MBF_PresetSingleRgs FB ……	2-24
MBF_PresetSingleRgsAns FB ……	2-25

MBF_FetchCommEvnCntr FB	2-26
MBF_FetchCommEvnCntAns FB	2-27
MBF_ForceMitCoil FB	2-28
MBF_ForceMitCoilAns FB	2-29
MBF_PresetMitRgs FB	2-30
MBF_PresetMitRgsAns FB	2-31
2-4 エラーコード	2-32

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。

- ・ MULTIPROG への Modbus 機能の登録方法
- ・ PLC プログラミング用 Modbus 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 Modbus ライブラリ

本章では PHOENIX CONTACT 社製 MULTIPROG における Modbus 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。

1-2 Modbus とは

Modbus とは、Modicon 社が同社向け PLC に策定したシリアル通信プロトコルになります。

現在では、産業用機器を接続する際の一般的な手段となっている通信方式の1つです。

シリアル通信プロトコルである為、弊社製の産業用 PC/産業用パネル PC/オールインワンコントローラに実装されているシリアルポートを使用します。

機種ごとにシリアルポート数は異なりますが、最大で2ポートまでサポートしています。

シリアルポートは、RS-232C 固定ポートと、RS-232C、RS422/485 に切り替え可能なポートが用意されています。

1-3 基本設定

弊社ソフトウェア PLC でシリアルポートを使用するには、実行環境（弊社製産業用 PC/産業用パネル PC/オールインワンコントローラ）側の設定が必要になります。

本項では、必要となる 3 種類の設定について説明します。

1-3-1 ハードウェア設定

弊社産業用 PC には、シリアルポートの機能設定を行うための DipSW があります。

DipSW の設定は

Mode 設定スイッチ

SIO ポート設定スイッチ

の 2 つの設定が必要です。

Mode 設定スイッチでは、ご使用になられるポートの通信仕様設定 (OFF : RS232C / ON : RS-422/485) を変更できます。

SIO ポート設定スイッチでは、全二重/半二重、TX/RX の終端設定を行います。

設定変更の SW は機種により異なる事がありますので、詳細については「産業用 PC/産業用パネル PC/オールインワンコントローラ ユーザーズマニュアル」を参照してください。

*)上記スイッチは弊社産業用パネル PC である AP を例にしています。

1-3-2 ソフトウェア設定

ハードウェア設定にあわせて、Windows ソフトウェアの設定が必要になります。

コントロールパネル内にインストールされている、ASD Config ツールを使用して、シリアルポートの設定を行います。

設定内容は、

ポート毎の通信仕様 (RS-232C/RS-422/RS-485)

RS-485 使用時の送信 Enable 時間

になります。

詳細については、弊社産業用 PC/産業用パネル PC/オールインワンコントローラに付属のソフトウェアマニュアルを参照してください。

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_FwLib_Modbus」になります。

2-1 機能概要

1) Modbus ファンクションブロック

表 2-1-1. Modbus ファンクションブロック一覧

ファンクション ブロック名	機 能
MBF_CommOpen	Modbus 通信をシリアル通信モードでオープンします。
MBF_CommClose	Modbus 通信をクローズします。
MBF_ReadCoilStatus	Modbus スレーブからコイル情報の取得を開始します。
MBF_ReadCoilStatusAns	MBF_ReadCoilStatus 関数の処理完了を確認します。
MBF_ReadInputStatus	Modbus スレーブから入力ステータス情報の取得を開始します。
MBF_ReadInputStatusAns	MBF_ReadInputStatus 関数の処理完了を確認します。
MBF_ReadHoldingRgs	Modbus スレーブから保持レジスタ情報の取得を開始します。
MBF_ReadHoldingRgsAns	MBF_ReadHoldingRgs 関数の処理完了を確認します。
MBF_ReadInputRgs	Modbus スレーブから入力レジスタ情報の取得を開始します。
MBF_ReadInputRgsAns	MBF_ReadInputRgs 関数の処理完了を確認します。
MBF_ForceSingleCoil	Modbus スレーブへ1コイルデータの書き込みを開始します。
MBF_ForceSingleCoilAns	MBF_ForceSingleCoil 関数の処理完了を確認します。
MBF_PresetSingleRgs	Modbus スレーブへ1保持データの書き込みを開始します。
MBF_PresetSingleRgsAns	MBF_PresetSingleRgs 関数の処理完了を確認します。
MBF_FetchCommEvnCntr	Modbus スレーブのイベントカウンタ情報を取得します。
MBF_FetchCommEvnCntAns	MBF_FetchCommEvnCntr 関数の処理完了を確認します。
MBF_ForceMltCoil	Modbus スレーブへ複数のコイル情報の書き込みを開始します。
MBF_ForceMltCoilAns	MBF_ForceMltCoil 関数の処理完了を確認します。
MBF_PresetMltRgs	Modbus スレーブから保持レジスタ情報の取得を開始します。
MBF_PresetMltRgsAns	MBF_PresetMltRgs 関数の処理完了を確認します。

2) ライブラリフローチャート

ライブラリを使用する際のファンクションブロック呼び出しのフローチャートを以下に示します。

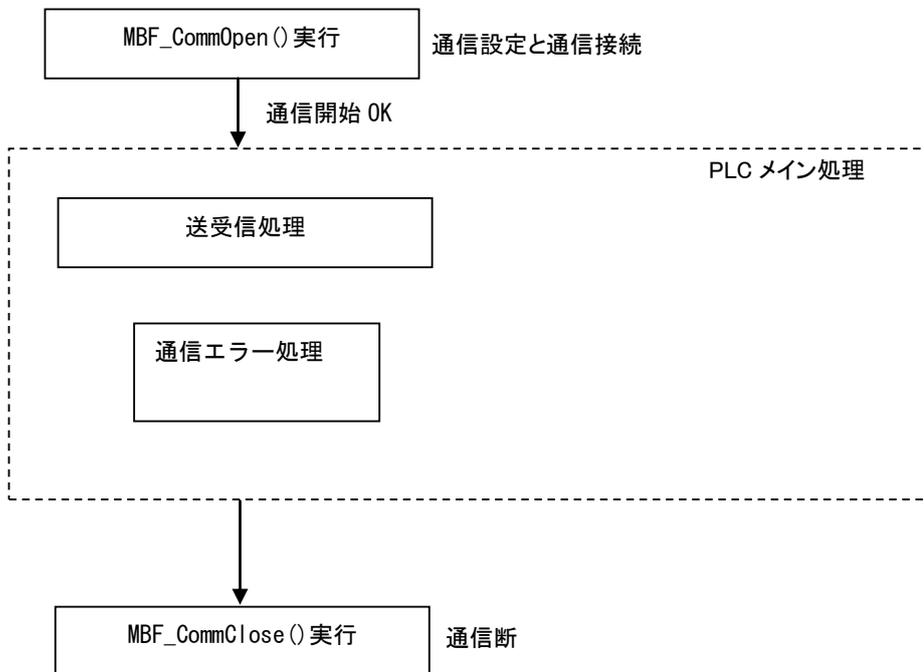


図 2-1-1. ライブラリフローチャート

PLC 開始後、MBF_CommOpen を実行し通信設定・接続を行うことでシリアル通信が可能になります。通信が正常に開始されれば、MBF_Read...などのファンクションブロックを使用し、データの送受信を行う事が出来ます。PLCの停止時には、自動的にMBF_CommCloseが実行されますが、PLC開発中などの動作確認時には、MBF_CommCloseを実行するようにしてください。

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「C:\Program Files\PHOENIX CONTACT Software\MULTIPROG X.XX Build XXX\plc\FW_LIB」になります。

*)Xの数字はVerによって異なります。

- ①MULTIPROG 画面の左ペインにある「ライブラリ」を選択します。

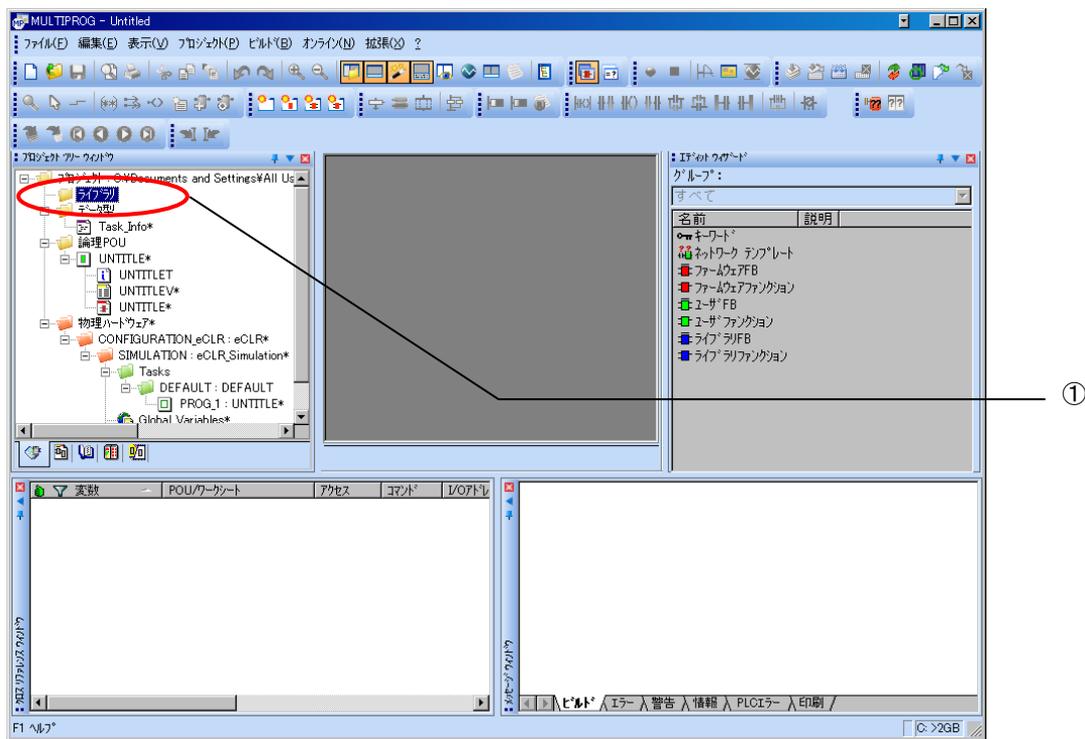


図 2-2-1. ライブラリ選択画面

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。

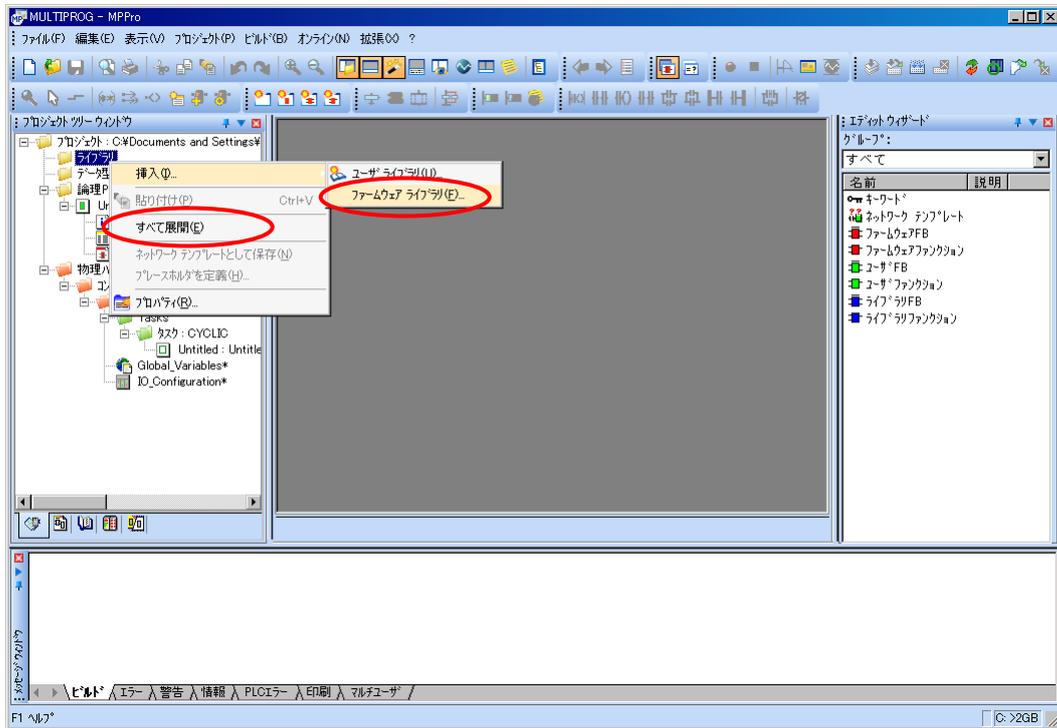


図 2-2-2. ライブラリ挿入画面①

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。

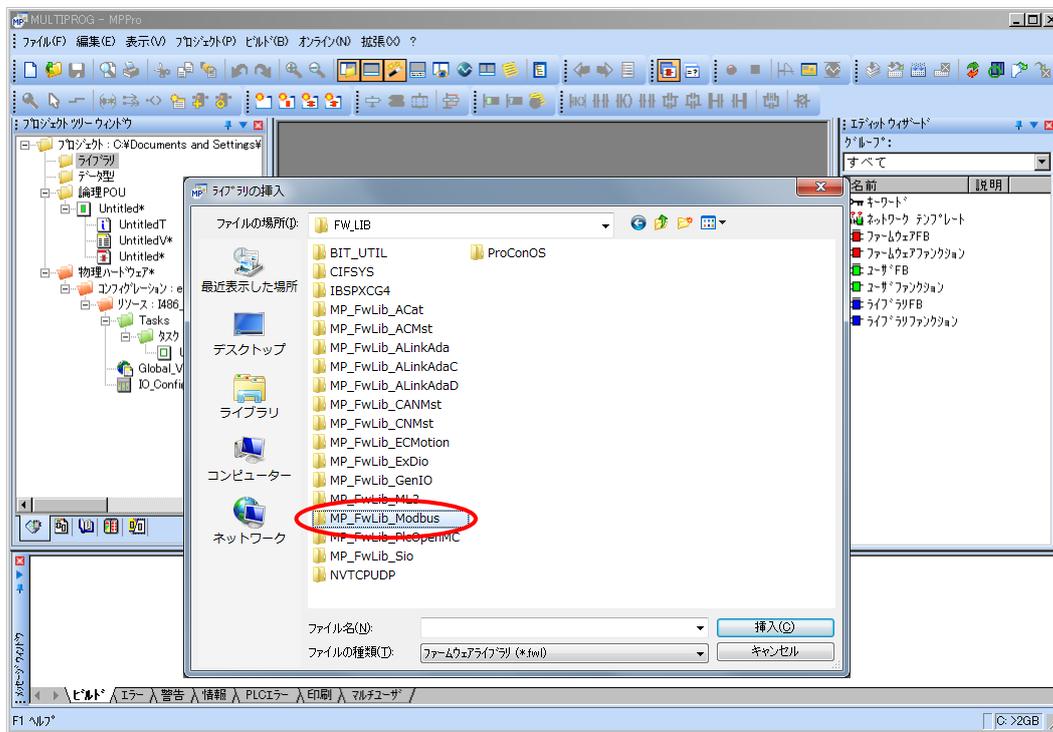


図 2-2-3. ライブラリ挿入画面②

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。

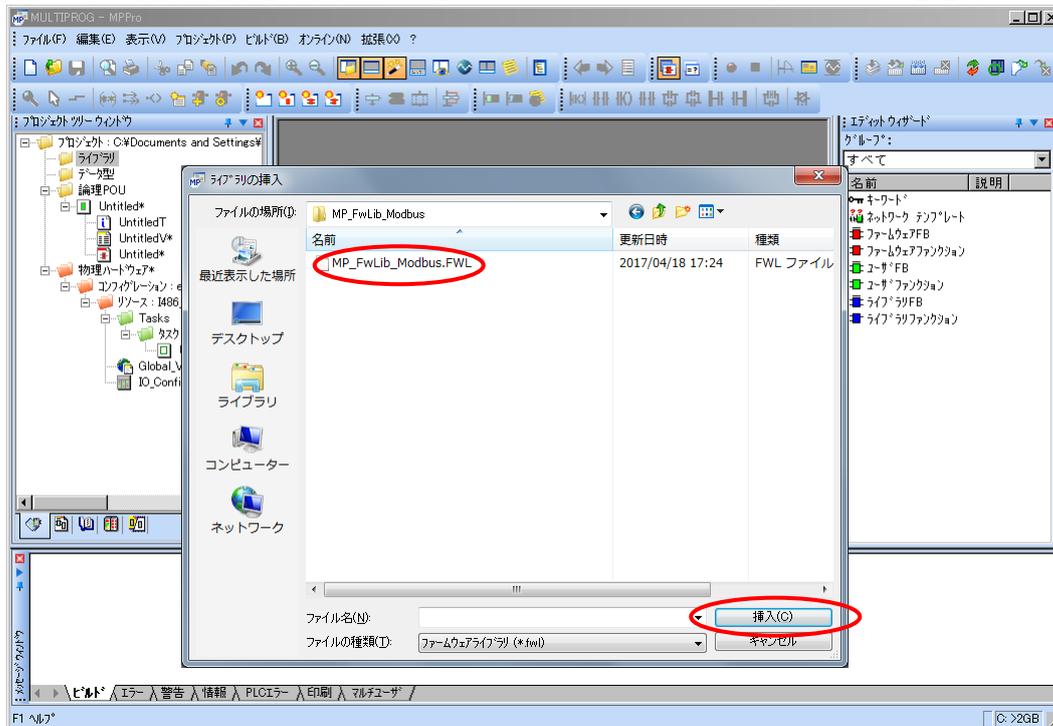


図 2-2-4. ライブラリ挿入画面③

- 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。

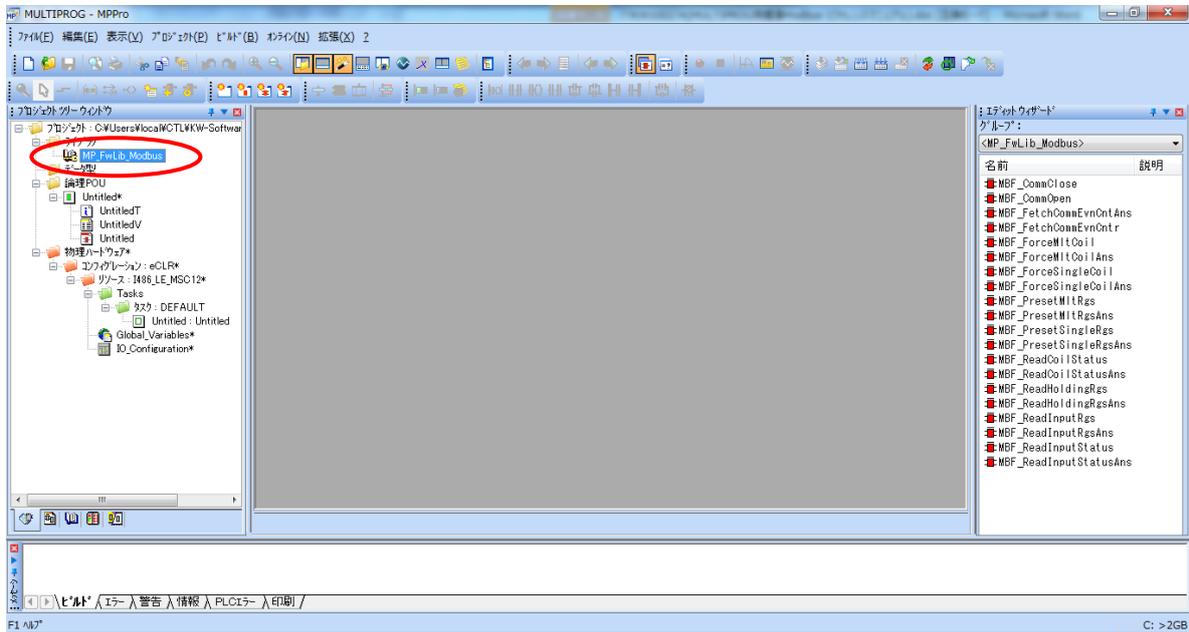


図 2-2-5. ライブラリ挿入画面④

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 Modbus 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コマンドを実行するための入力として「Execute」、コマンドの応答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。

Execute

立ち上がりエッジによりコマンドを実行します。以降は応答待ち状態になります。

Done 出力が True となる前に本パラメータを False にする事で応答待ちを停止し、全ての出力を初期化します。応答確認による終了確認は処理されなくなりますが、コマンドの実行が可能な状態であれば実行されます。コマンドのキャンセル処理ではありません。

ファンクションブロックの入力パラメータを変更した場合は、この入力を False→True と変化させてコマンドを再度実行してください。

Done

ファンクションブロックが正常終了の場合、本出力は TRUE となり処理の完了を通知します。Error と同時に TRUE になる事はありません。

Error

ファンクションブロックが異常終了の場合、本出力が TRUE となり処理の異常を通知します。Done と同時に TRUE になる事はありません。Error=TRUE の応答があった場合は、ErrorID 出力パラメータを確認してください。

ANY データ型

いくつかの読み込み用ファンクションブロックには、読み込みデータの格納用バッファとして ANY 型の Read という入出力が用意されています。

Read に接続する変数の型はあらかじめ定義しておく必要があり、PLC のプロジェクトに以下のデータ型を追加して使用してください。

```
TYPE
    BUFFER : ARRAY [1..XXXXX] OF BYTE;
END_TYPE
または、
TYPE
    BUFFER2 : ARRAY [1..XXXXX] OF WORD;
END_TYPE
```

ここで BUFFER/ BUFFER2 は任意の型名を命名してください。
XXXXX の部分にはバッファサイズを指定してください。

FuncID

実行するファンクションブロックを管理するための ID。例えば MBF_ReadCoilStatus と MBF_ReadCoilStatusAns のような対になるファンクションブロックでは同じ FuncID を使用してください。

FuncID の使用方法
① 正常な場合

MBF_ReadCoilStatus (FuncID=1) : Execute =TRUE で実行
出力側 Done=TRUE で完了
MBF_ReadCoilStatusAns (FuncID=1) : Execute =TRUE で実行
出力側 Done=TRUE で完了
これで MBF_ReadCoilStatusAns の Read は正常な結果が読み出せる。

② 異常となる場合 (FuncID とファンクションブロックを間違えた場合)

MBF_ReadInputStatus (FuncID=1) : Execute =TRUE で実行
出力側 Done=TRUE で完了
MBF_ReadCoilStatusAns (FuncID=1) : Execute =TRUE で実行
出力側 Done=FALSE、Error=TRUE で完了
MBF_ReadCoilStatusAns はエラーとなり ErrorID を確認するとエラー内容が分かる。
この場合、「MBM_ER_MODBUS_FUNC (エラーコード：0200h)」となります。

③ 異常となる場合 (Ans 側のファンクションブロックを先に呼び出した場合)

MBF_ReadCoilStatusAns (FuncID=1) : Execute =TRUE で実行
出力側 Done=FALSE、Error=TRUE で完了
MBF_ReadCoilStatusAns はエラーとなり ErrorID を確認するとエラー内容が分かる。
この場合、「MBM_ER_NOTINIT (エラーコード：0002h)」となります。

④ 補足説明：
正常な場合 (複数：呼び出す順番を入れ替えても正常な応答となる)

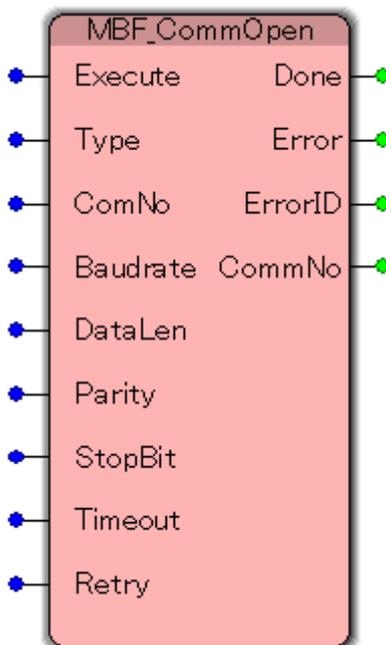
MBF_ReadCoilStatus (FuncID=1) : Execute =TRUE で実行
出力側 Done=TRUE で完了
MBF_ReadCoilStatus (FuncID=2) : Execute =TRUE で実行
出力側 Done=TRUE で完了
MBF_ReadCoilStatusAns (FuncID=1) : Execute =TRUE で実行
出力側 Done=TRUE で完了
MBF_ReadCoilStatusAns の Read は正常な結果が読み出せる。
MBF_ReadCoilStatusAns (FuncID=2) : Execute =TRUE で実行
出力側 Done=TRUE で完了
MBF_ReadCoilStatusAns の Read は正常な結果が読み出せる。

MBF_CommOpen FB

機能

Modbus 通信をシリアル通信モードで開始します。

書式



入力

BOOL	Execute	:	実行フラグ (立ち上りエッジにより実行)
INT	Type	:	Modbus 通信タイプ (0 : ASCII モード 1 : RTU モード)
INT	ComNo	:	シリアルポート番号
INT	Baudrate	:	通信速度
INT	DataLen	:	データ長
INT	Parity	:	パリティ
INT	StopBit	:	ストップビット
INT	Timeout	:	タイムアウト時間設定 (単位 msec, 0 はタイムアウトしない)
INT	Retry	:	リトライ回数設定

出力

BOOL	Done	:	正常完了フラグ (0 : 応答なし, 1 : 実行完了)
BOOL	Error	:	エラーフラグ (0 : エラーなし, 1 : エラーあり)
DWORD	ErrorID	:	エラーコード (エラーコード一覧を参照)
INT	CommNo	:	通信番号 (0 から連番)

説明

Modbus 通信をシリアル通信で通信開始します。
 通信番号 (0 から連番) を CommNo に格納します。
 本 FB の正常完了後、Modbus を使った送受信が可能になります。本ライブラリを使用して Modbus 通信を行う前に、必ずコールする必要があります。

シリアル通信設定

Baudrate : 通信速度

通信速度	内容
0	1200 bps
1	2400 bps
2	4800 bps
3	9600 bps
4	19200 bps
5	38400 bps
6	57600 bps
7	115200 bps

DataLen : データビット長

データビット長	内容
0	7 ビット
1	8 ビット

Parity : パリティビット

パリティビット	内容
0	NON : ノンパリティ
1	ODD : 奇数パリティ
2	EVEN : 偶数パリティ

StopBit : ストップビット長

ストップビット長	内容
0	1 ビット
1	1.5 ビット
2	2 ビット

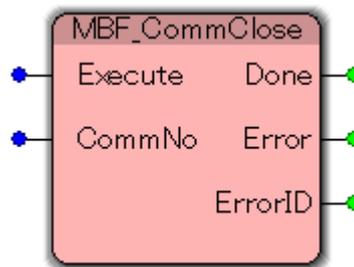
Timeout : タイムアウト時間
[0 ~ 65535] 0:タイムアウトしない

Retry : リトライ回数
[0 ~ 65535]

MBF_CommClose FB

機能 Modbus 通信を終了します。

書式



入力

BOOL	Execute	:	実行フラグ (立ち上りエッジにより実行)
INT	CommNo	:	通信番号

出力

BOOL	Done	:	正常完了フラグ (0 : 応答なし, 1 : 実行完了)
BOOL	Error	:	エラーフラグ (0 : エラーなし, 1 : エラーあり)
DWORD	ErrorID	:	エラーコード (エラーコード一覧を参照)

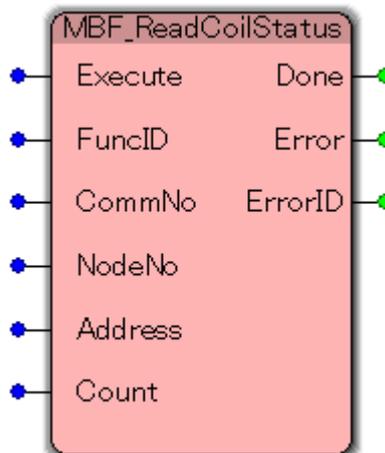
説明 Modbus 通信を切断します。
 本 FB をコール後は、MBF_CommOpen 以外の FB の実行は不可になります。
 本ライブラリの使用を終了する場合は、必ずコールする必要があります。

MBF_ReadCoilStatus FB

機能

Modbus スレーブにコイル情報の取得を指令します。

書式



入力

BOOL	Execute	:	実行フラグ (立ち上りエッジにより実行)
INT	FuncID	:	関数 ID
INT	CommNo	:	通信番号
INT	NodeNo	:	Modbus ノード番号
UDINT	Address	:	コイル読み出し開始アドレス
UDINT	Count	:	読み出しコイル数 (bit 単位)

出力

BOOL	Done	:	正常完了フラグ (0 : 応答なし, 1 : 実行完了)
BOOL	Error	:	エラーフラグ (0 : エラーなし, 1 : エラーあり)
DWORD	ErrorID	:	エラーコード (エラーコード一覧を参照)

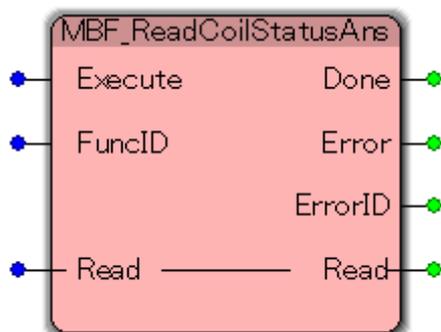
説明

Modbus スレーブからコイル情報読み出しを開始します。
 実行処理を監視するために関数 ID を指定します。
 処理完了は MBF_ReadCoilStatusAns 関数に、同じ関数 ID を指定することで確認します。

MBF_ReadCoilStatusAns FB

機能 MBF_ReadCoilStatus 関数の処理完了を確認します。

書式



入力 BOOL Execute : 実行フラグ (立ち上りエッジにより実行)
 INT FuncID : 関数 ID

出力 BOOL Done : 正常完了フラグ (0 : 応答なし, 1 : 実行完了)
 BOOL Error : エラーフラグ (0 : エラーなし, 1 : エラーあり)
 DWORD ErrorID : エラーコード (エラーコード一覧を参照)

入出力 ANY Read : コイル情報
 (2-3 ファンクションブロックリファレンスの Any データ型を参照)

説明 MBF_ReadCoilStatus 関数の処理完了を確認します。
 正常終了の場合、Read に読み出したデータが格納されます。
 読み出しバイト数は読み出しコイル数/8 です。

読出しコイル数の設定についての説明

読出しコイル数が 10 の場合

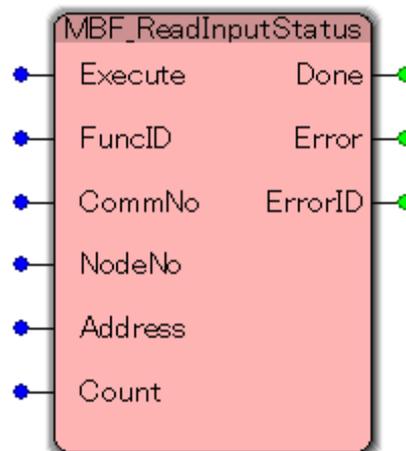
1バイト	8	7	6	5	4	3	2	1
2バイト	リザーブ						10	9

MBF_ReadInputStatus FB

機能

Modbus スレーブから入力ステータス情報の取得を開始します。

書式



入力

BOOL	Execute	:	実行フラグ (立ち上りエッジにより実行)
INT	FuncID	:	関数 ID
INT	CommNo	:	通信番号
INT	NodeNo	:	Modbus ノード番号
UDINT	Address	:	インプットステータス読み出し開始アドレス
UDINT	Count	:	読み出しインプットステータス数 (Bit 単位)

出力

BOOL	Done	:	正常完了フラグ (0 : 応答なし, 1 : 実行完了)
BOOL	Error	:	エラーフラグ (0 : エラーなし, 1 : エラーあり)
DWORD	ErrorID	:	エラーコード (エラーコード一覧を参照)

説明

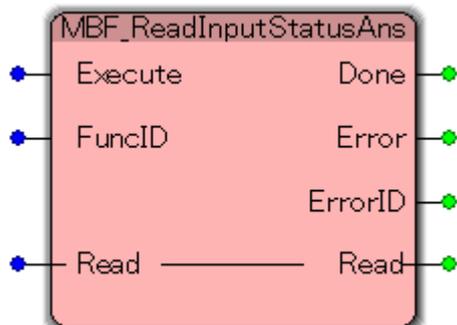
Modbus スレーブからインプットステータス情報読み出しを開始します。
 実行処理を監視するために関数 ID を指定します。
 処理完了は MBF_ReadInputStatusAns 関数に、同じ関数 ID を指定することで確認します。

MBF_ReadInputStatusAns FB

機能

MBF_ReadInputStatus 関数の処理完了を確認します。

書式



入力

BOOL Execute : 実行フラグ (立ち上りエッジにより実行)
 INT FuncID : 関数 ID

出力

BOOL Done : 正常完了フラグ (0 : 応答なし, 1 : 実行完了)
 BOOL Error : エラーフラグ (0 : エラーなし, 1 : エラーあり)
 DWORD ErrorID : エラーコード (エラーコード一覧を参照)

入出力

ANY Read : 入力ステータス情報
 (2-3 ファンクションブロックリファレンスの Any データ型を参照)

説明

MBF_ReadInputStatus 関数の処理完了を確認します。
 正常終了の場合は、Read に読み出したデータが格納されます。
 読み出しバイト数は読み出しインプットステータス数/8 です。

読み出しインプットステータス数の設定についての説明
 読み出しインプットステータス数が 10 の場合

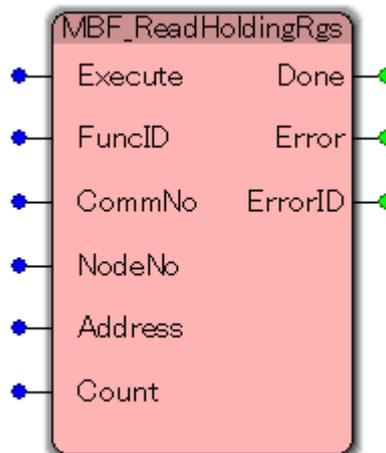
1 バイト	8	7	6	5	4	3	2	1
2 バイト	リザーブ						10	9

MBF_ReadHoldingRgs FB

機能

Modbus スレーブから保持レジスタ情報の取得を開始します。

書式



入力

BOOL	Execute	:	実行フラグ (立ち上りエッジにより実行)
INT	FuncID	:	関数 ID
INT	CommNo	:	通信番号
INT	NodeNo	:	Modbus ノード番号
UDINT	Address	:	保持レジスタ読み出し開始アドレス
UDINT	Count	:	読み出し保持レジスタ数 (Word 単位)

出力

BOOL	Done	:	正常完了フラグ (0 : 応答なし, 1 : 実行完了)
BOOL	Error	:	エラーフラグ (0 : エラーなし, 1 : エラーあり)
DWORD	ErrorID	:	エラーコード (エラーコード一覧を参照)

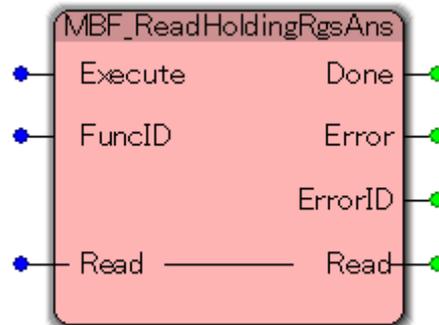
説明

Modbus スレーブから保持レジスタ読み出しを開始します。
 実行処理を監視するために関数 ID を指定します。
 処理完了は MBF_ReadHoldingRgsAns 関数に、同じ関数 ID を指定することで確認します。

MBF_ReadHoldingRgsAns FB

機能

MBF_ReadHoldingRgs 関数の処理完了を確認します。

書式**入力**

BOOL Execute : 実行フラグ (立ち上りエッジにより実行)
 INT FuncID : 関数 ID

出力

BOOL Done : 正常完了フラグ (0 : 応答なし, 1 : 実行完了)
 BOOL Error : エラーフラグ (0 : エラーなし, 1 : エラーあり)
 DWORD ErrorID : エラーコード (エラーコード一覧を参照)

入出力

ANY Read : 保持レジスタ情報
 (2-3 ファンクションブロックリファレンスの Any データ型を参照)

説明

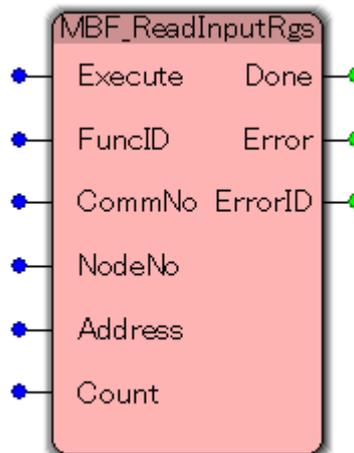
MBF_ReadHoldingRgs 関数の処理完了を確認します。
 正常終了の場合は、Read に読み出したデータが格納されます。
 読み出すデータは WORD 単位の配列になります。

MBF_ReadInputRgs FB

機能

Modbus スレーブから入力レジスタ情報の取得を開始します。

書式



入力

BOOL	Execute	:	実行フラグ (立ち上りエッジにより実行)
INT	FuncID	:	関数 ID
INT	CommNo	:	通信番号
INT	NodeNo	:	Modbus ノード番号
UDINT	Address	:	入力レジスタ読み出し開始アドレス
UDINT	Count	:	読み出し入力レジスタ数 (Word 単位)

出力

BOOL	Done	:	正常完了フラグ (0 : 応答なし, 1 : 実行完了)
BOOL	Error	:	エラーフラグ (0 : エラーなし, 1 : エラーあり)
DWORD	ErrorID	:	エラーコード (エラーコード一覧を参照)

説明

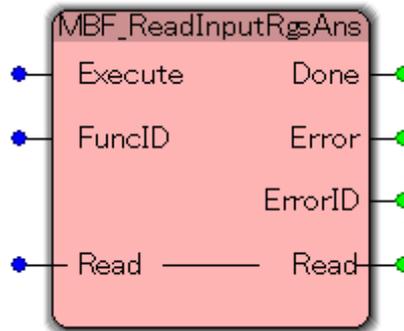
Modbus スレーブから入力レジスタ読み出しを開始します。
 実行処理を監視するために関数 ID を指定します。
 処理完了は MBF_ReadInputRgsAns 関数に、同じ関数 ID を指定することで確認します。

MBF_ReadInputRgsAns FB

機能

MBF_ReadInputRgs 関数の処理完了を確認します。

書式



入力

BOOL Execute : 実行フラグ (立ち上りエッジにより実行)
 INT FuncID : 関数 ID

出力

BOOL Done : 正常完了フラグ (0 : 応答なし, 1 : 実行完了)
 BOOL Error : エラーフラグ (0 : エラーなし, 1 : エラーあり)
 DWORD ErrorID : エラーコード (エラーコード一覧を参照)

入出力

ANY Read : 入力レジスタ情報
 (2-3 ファンクションブロックリファレンスの Any データ型を参照)

説明

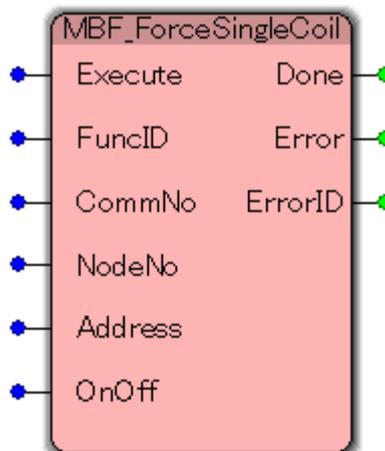
MBF_ReadInputRgs 関数の処理完了を確認します。
 正常終了の場合は、Read に読み出したデータが格納されます。
 読み出すデータは WORD 単位の配列になります。

MBF_ForceSingleCoil FB

機能

Modbus スレーブへ 1 コイルデータの書き込みを開始します。

書式



入力

BOOL	Execute	:	実行フラグ (立ち上りエッジにより実行)
INT	FuncID	:	関数 ID
INT	CommNo	:	通信番号
INT	NodeNo	:	Modbus ノード番号
UDINT	Address	:	コイル書き込みアドレス
BOOL	OnOff	:	書き込みデータ (0 : OFF 1 : ON)

出力

BOOL	Done	:	正常完了フラグ (0 : 応答なし, 1 : 実行完了)
BOOL	Error	:	エラーフラグ (0 : エラーなし, 1 : エラーあり)
DWORD	ErrorID	:	エラーコード (エラーコード一覧を参照)

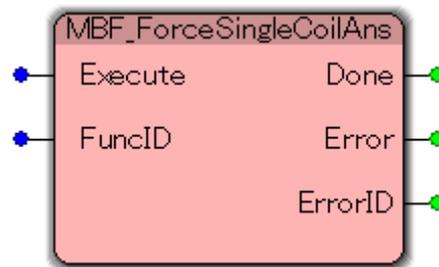
説明

Modbus スレーブへコイルデータ 1bit の書き込みを開始します。
 実行処理を監視するために関数 ID を指定します。
 処理完了は MBF_ForceSingleCoilAns 関数に、同じ関数 ID を指定することで確認します。

MBF_ForceSingleCoilAns FB

機能 MBF_ForceSingleCoil 関数の処理完了を確認します。

書式



入力

BOOL	Execute	:	実行フラグ (立ち上りエッジにより実行)
INT	FuncID	:	関数 ID

出力

BOOL	Done	:	正常完了フラグ (0 : 応答なし, 1 : 実行完了)
BOOL	Error	:	エラーフラグ (0 : エラーなし, 1 : エラーあり)
DWORD	ErrorID	:	エラーコード (エラーコード一覧を参照)

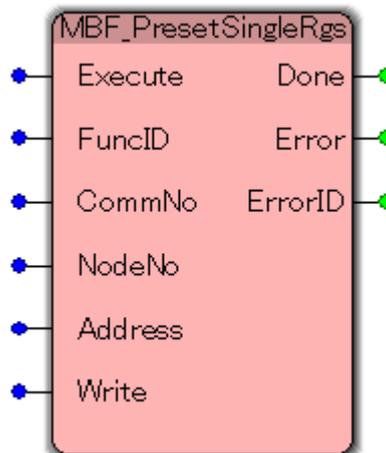
説明 MBF_ForceSingleCoil 関数の処理完了を確認します。

MBF_PresetSingleRgs FB

機能

Modbus スレーブへ 1 保持データの書き込みを開始します。

書式



入力

BOOL	Execute	:	実行フラグ (立ち上りエッジにより実行)
INT	FuncID	:	関数 ID
INT	CommNo	:	通信番号
INT	NodeNo	:	Modbus ノード番号
UDINT	Address	:	保持レジスタ書き込みアドレス
WORD	Write	:	書き込みワードデータ

出力

BOOL	Done	:	正常完了フラグ (0 : 応答なし, 1 : 実行完了)
BOOL	Error	:	エラーフラグ (0 : エラーなし, 1 : エラーあり)
DWORD	ErrorID	:	エラーコード (エラーコード一覧を参照)

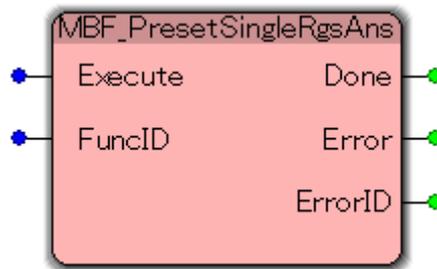
説明

Modbus スレーブへ保持レジスタ 1 ワード分の書き込みを開始します。
 実行処理を監視するために関数 ID を指定します。
 処理完了は MBF_PresetSingleRgsAns 関数に、同じ関数 ID を指定することで確認します。

MBF_PresetSingleRgsAns FB

機能 MBF_PresetSingleRgs 関数の処理完了を確認します。

書式



入力

BOOL	Execute	:	実行フラグ (立ち上りエッジにより実行)
INT	FuncID	:	関数 ID

出力

BOOL	Done	:	正常完了フラグ (0 : 応答なし, 1 : 実行完了)
BOOL	Error	:	エラーフラグ (0 : エラーなし, 1 : エラーあり)
DWORD	ErrorID	:	エラーコード (エラーコード一覧を参照)

説明 MBF_PresetSingleRgs 関数の処理完了を確認します。

MBF_FetchCommEvnCntr FB

機能

Modbus スレーブのイベントカウンタ情報を取得します。

書式



入力

BOOL	Execute	:	実行フラグ (立ち上りエッジにより実行)
INT	FuncID	:	関数 ID
INT	CommNo	:	通信番号
INT	NodeNo	:	Modbus ノード番号

出力

BOOL	Done	:	正常完了フラグ (0 : 応答なし, 1 : 実行完了)
BOOL	Error	:	エラーフラグ (0 : エラーなし, 1 : エラーあり)
DWORD	ErrorID	:	エラーコード (エラーコード一覧を参照)

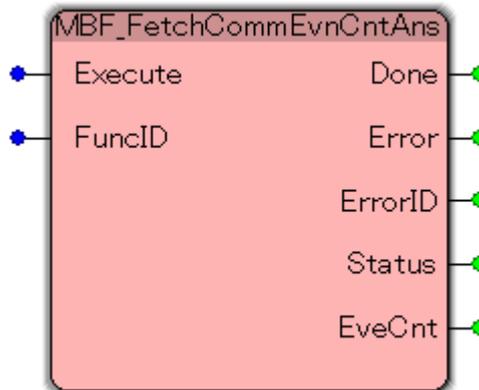
説明

Modbus スレーブのイベント情報を取得します。
 実行処理を監視するために関数 ID を指定します。
 処理完了は MBF_FetchCommEvnCntAns 関数に、同じ関数 ID を指定することで確認します。

MBF_FetchCommEvnCntAns FB

機能 MBF_FetchCommEvnCntAns 関数の処理完了を確認します。

書式



入力

BOOL	Execute	:	実行フラグ (立ち上りエッジにより実行)
INT	FuncID	:	関数 ID

出力

BOOL	Done	:	正常完了フラグ (0 : 応答なし, 1 : 実行完了)
BOOL	Error	:	エラーフラグ (0 : エラーなし, 1 : エラーあり)
DWORD	ErrorID	:	エラーコード (エラーコード一覧を参照)
WORD	Status	:	ステータス情報
WORD	EveCnt	:	イベントカウンタ情報

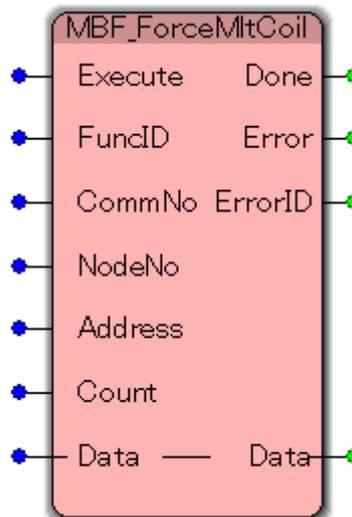
説明 MBF_FetchCommEvnCntAns 関数の処理完了を確認します。
 正常終了した場合は、Status にステータス情報が、EveCnt にイベントカウンタ情報が格納されます。

MBF_ForceMltCoil FB

機能

Modbus スレーブへ複数のコイル情報の書き込みを開始します。

書式



入力

- BOOL Execute : 実行フラグ (立ち上りエッジにより実行)
- INT FuncID : 関数 ID
- INT CommNo : 通信番号
- INT NodeNo : Modbus ノード番号
- UDINT Address : コイル書き込み開始アドレス
- UDINT Count : 書き込みコイル数 (Bit 単位)

出力

- BOOL Done : 正常完了フラグ (0 : 応答なし, 1 : 実行完了)
- BOOL Error : エラーフラグ (0 : エラーなし, 1 : エラーあり)
- DWORD ErrorID : エラーコード (エラーコード一覧を参照)

入出力

- ANY Data : コイル書き込みデータ (1Byte 毎に 8bit 分のコイルデータを格納) (2-3 ファンクションブロックリファレンスの Any データ型を参照)

説明

Modbus スレーブへコイルデータ複数 bit 分の書き込みを開始します。
 実行処理を監視するために関数 ID を指定します。
 処理完了は MBF_ForceMltCoilAns 関数に、同じ関数 ID を指定することで確認します。

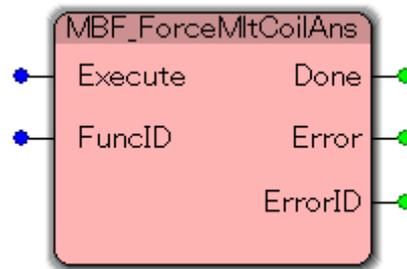
書き込みコイル数の設定についての説明
 書き込みコイル数が 10 の場合

1 バイト	8	7	6	5	4	3	2	1
2 バイト	リザーブ						10	9

MBF_ForceMltCoilAns FB

機能 MBF_ForceMltCoil 関数の処理完了を確認します。

書式



入力

BOOL	Execute	:	実行フラグ (立ち上りエッジにより実行)
INT	FuncID	:	関数 ID

出力

BOOL	Done	:	正常完了フラグ (0 : 応答なし, 1 : 実行完了)
BOOL	Error	:	エラーフラグ (0 : エラーなし, 1 : エラーあり)
DWORD	ErrorID	:	エラーコード (エラーコード一覧を参照)

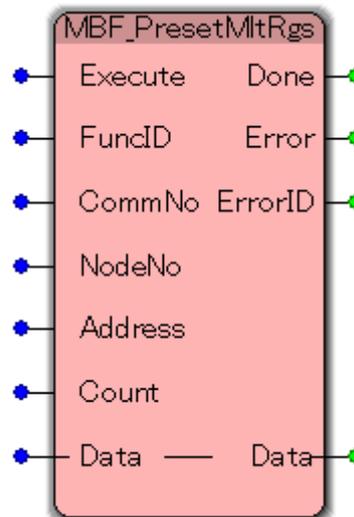
説明 MBF_ForceMltCoil 関数の処理完了を確認します。

MBF_PresetMltRgs FB

機能

Modbus スレーブから保持レジスタ情報の取得を開始します。

書式



入力

BOOL	Execute	:	実行フラグ (立ち上りエッジにより実行)
INT	FuncID	:	関数 ID
INT	CommNo	:	通信番号
INT	NodeNo	:	Modbus ノード番号
UDINT	Address	:	保持レジスタ書き込み開始アドレス
UDINT	Count	:	書き込み保持レジスタワード数

出力

BOOL	Done	:	正常完了フラグ (0 : 応答なし, 1 : 実行完了)
BOOL	Error	:	エラーフラグ (0 : エラーなし, 1 : エラーあり)
DWORD	ErrorID	:	エラーコード (エラーコード一覧を参照)

入出力

ANY	Data	:	保持レジスタ書き込みワードデータ (2-3 ファンクションブロックリファレンスの Any データ型を参照)
-----	------	---	--

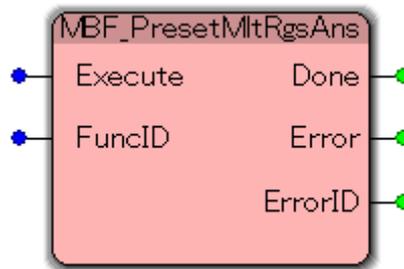
説明

Modbus スレーブへ保持レジスタ 1 ワード分の書き込みを開始します。
 実行処理を監視するために関数 ID を指定します。
 処理完了は MBF_PresetMltRgsAns 関数に、同じ関数 ID を指定することで確認します。
 書き込みデータは WORD 単位の配列になります。

MBF_PresetMltRgsAns FB

機能 MBF_PresetMltRgs 関数の処理完了を確認します。

書式



入力

BOOL	Execute	:	実行フラグ (立ち上りエッジにより実行)
INT	FuncID	:	関数 ID

出力

BOOL	Done	:	正常完了フラグ (0 : 応答なし, 1 : 実行完了)
BOOL	Error	:	エラーフラグ (0 : エラーなし, 1 : エラーあり)
DWORD	ErrorID	:	エラーコード (エラーコード一覧を参照)

説明 MBF_PresetMltRgs 関数の処理完了を確認します。

2-4 エラーコード

エラーコード一覧

表 2-4-1. エラーコード一覧

定義名	コード	内容
MBM_ER_OK	0000h	正常
MBM_ER_INVALID	0001h	無効な引数
MBM_ER_NOTINIT	0002h	初期化されていない
MBM_ER_ALREADYINIT	0003h	既に初期化されている
MBM_ER_OPENOVER	0004h	同時オープンできる数を超えた
MBM_ER_COMOPEN	0100h	シリアルポートオープン失敗
MBM_ER_COMSEND	0101h	シリアルポート送信失敗
MBM_ER_COMRECV	0102h	シリアルポート受信失敗
MBM_ER_COMSEND_TIMEOUT	0103h	シリアルポート送信タイムアウト
MBM_ER_COMRECV_TIMEOUT	0104h	シリアルポート受信タイムアウト
MBM_ER_COMRECV_FORMAT	0105h	シリアルポート送信データフォーマット異常
MBM_ER_MODBUS_FUNC	0200h	Modbus ファンクションコードエラー
MBM_ER_MODBUS_ADDR	0201h	Modbus アドレスエラー
MBM_ER_MODBUS_DATA	0202h	Modbus データエラー
MBM_ER_FUNC_FUNCID	0300h	無効な FuncID
MBM_ER_FUNC_RUNNING	0301h	実行中

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 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。