

リファレンスマニュアル

『MULTIPROG 用 標準 SIO』

# 目次

## はじめに

1) ……お願いと注意	1
-------------	---

## 第1章 MULTIPROG 用 標準 SIO ライブラリ

1-1 MULTIPROG とは	1-1
1-2 標準 SIO 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	2-1
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
ComOpen FB	2-8
ComClose FB	2-10
ComSend FB	2-11
ComRecv FB	2-12
ComGetRecvLen FB	2-14
ComGetStatus FB	2-15
ComClearStatus FB	2-16
ComPurge FB	2-17
ComEscFunc FB	2-18
ComSetStartCode FB	2-19
ComGetStartCode FB	2-20
ComSetEndCode FB	2-21
ComGetEndCode FB	2-22
2-4 エラーコード	2-23

## 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 への標準 SIO 機能の登録方法
- ・ PLC プログラミング用標準 SIO 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 標準 SIO ライブラリ

本章では PHOENIX CONTACT 社製 MULTIPROG における標準 SIO 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 標準 SIO とは

標準 SIO とは、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指します。

機種ごとにシリアルポート数は異なりますが、最大で 4 ポートまでサポートしています。

シリアルポートには、RS-232C 固定ポートと、RS-232C、RS422/485 に切り替え可能なポートがあります。

### 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。

それぞれのスイッチの設定と通信モードの組み合わせは表 1-3-1-1 のようになります。

表 1-3-1-1. 標準 SIO ファンクションブロッカー一覧

SIO ポート設定 SW MODE 設定 SW	全二重	半二重
OFF	RS-232C 通信	
ON	RS-424 通信	RS-485 通信

設定変更の SW の場所は機種により異なる事がありますので、詳細については各機種のユーザーズマニュアルまたは製品仕様書を参照してください。

#### 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、各機種のソフトウェアユーザーズマニュアルの『第 2 章 システムの操作』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\_Sio」になります。

### 2-1 機能概要

#### 1) 標準 SIO ファンクションブロック

表 2-1-1. 標準 SIO 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
ComOpen	通信設定と通信接続を行う。
ComClose	通信切断を行う。
ComSend	データ送信を行う。
ComRecv	データ受信を行う。
ComGetRecvLen	受信データ長を取得する。
ComGetStatus	通信状態を取得する。
ComClearStatus	通信状態のクリアを行う。
ComPurge	送信バッファ、受信バッファのクリアを行う。
ComEscFunc	通信拡張機能を行う。
ComSetStartCode	スタートコードを設定する。
ComGetStartCode	設定済みのスタートコードを取得する。
ComSetEndCode	エンドコードを設定する。
ComGetEndCode	設定済みのエンドコードを取得する。

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。

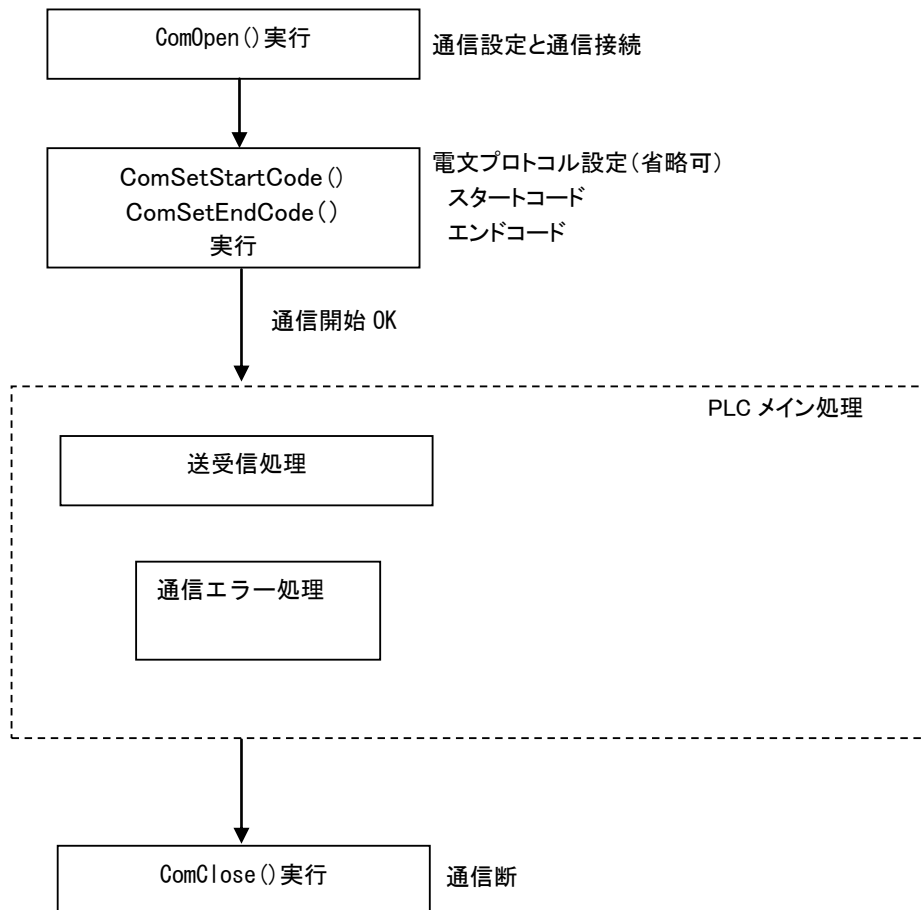


図 2-1-1. ライブラリフローチャート

PLC 開始後、ComOpen 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、ComRecv、ComSend を使用し電文の送受信を行う事が出来ます。PLC の停止時には、自動的に ComClose が実行されますが、PLC 開発中などの動作確認時には、ComClose 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。

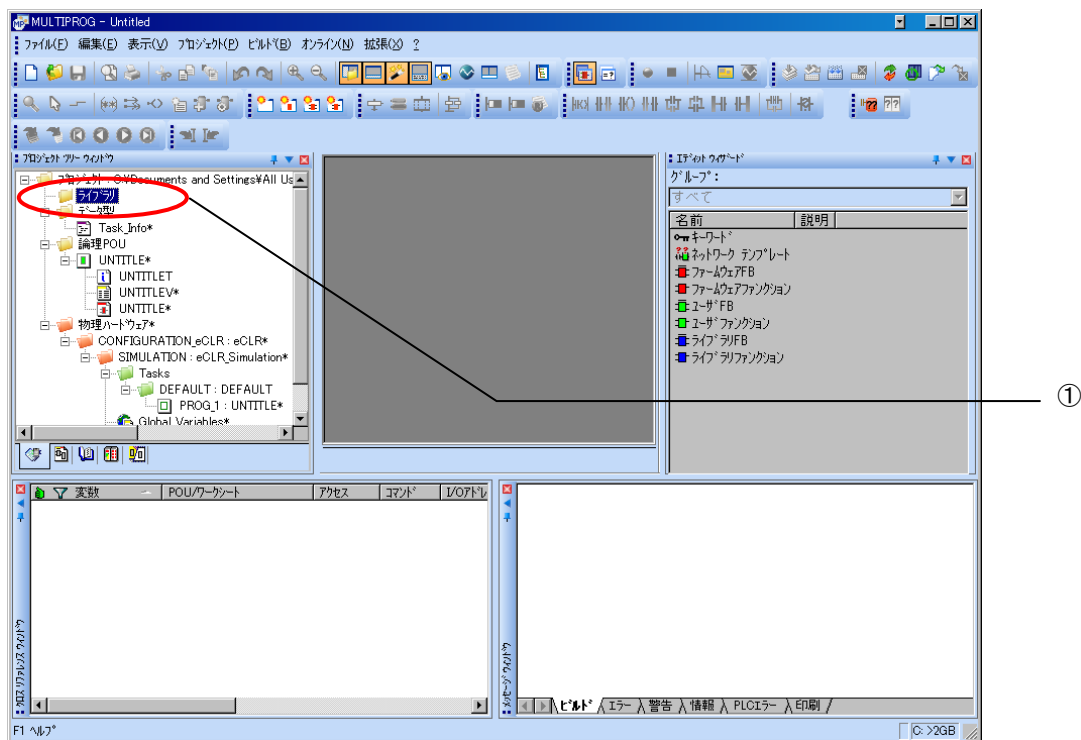


図 2-2-1. ライブラリ選択画面

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。

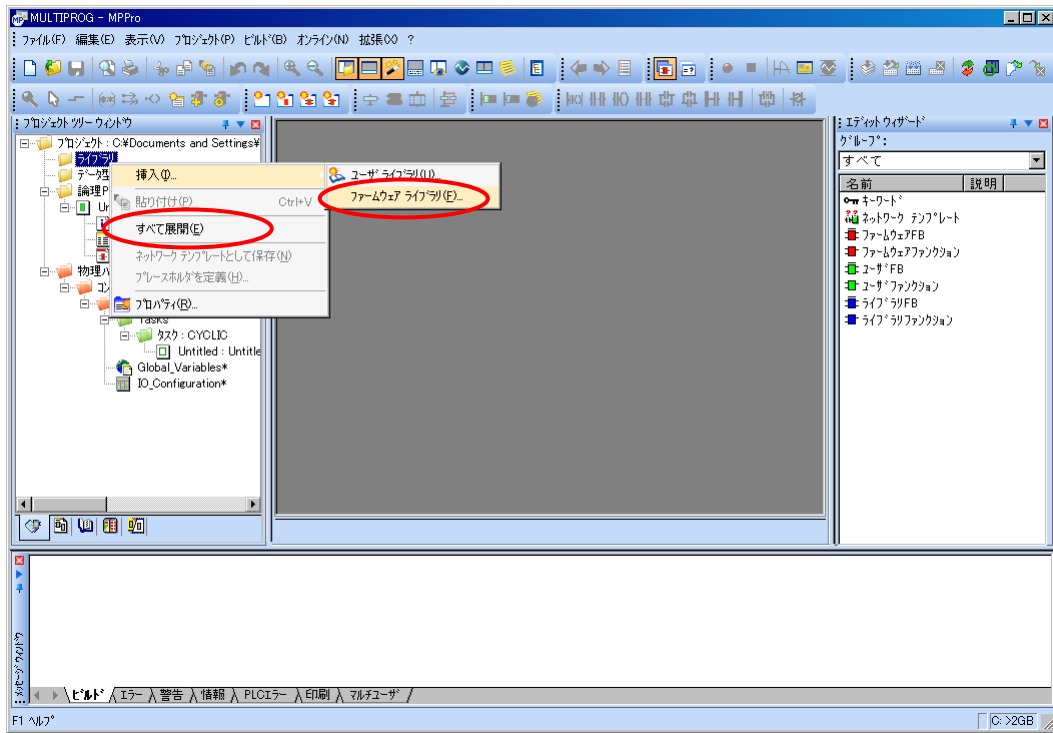


図 2-2-2. ライブラリ挿入画面①

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。

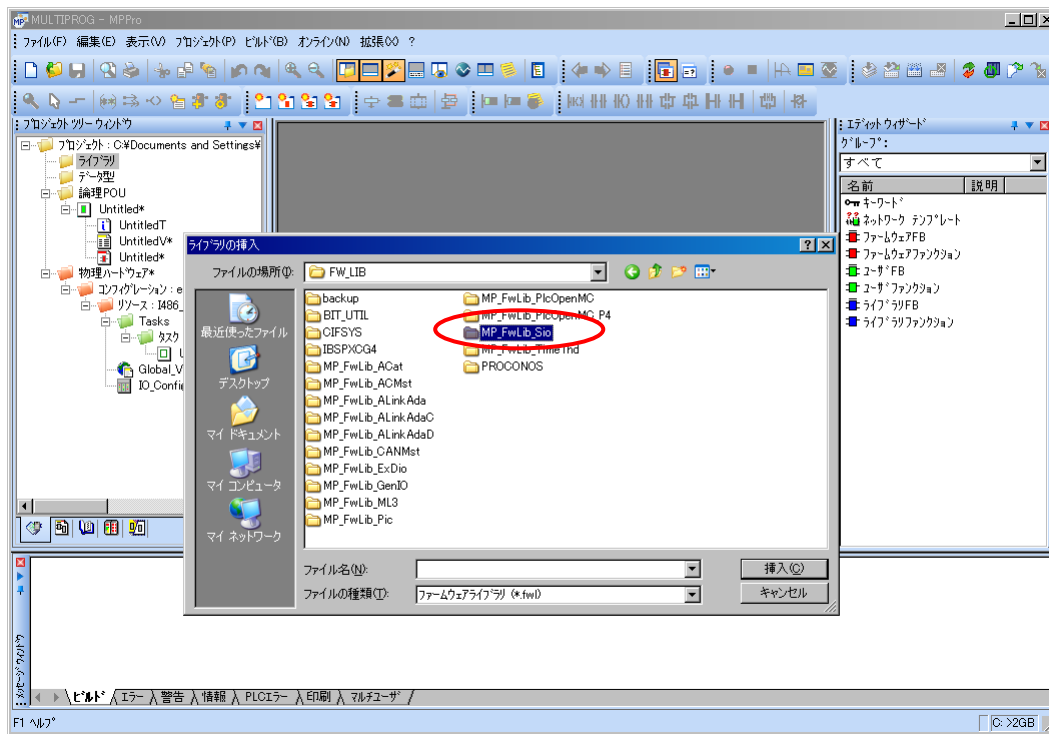


図 2-2-3. ライブラリ挿入画面②

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。

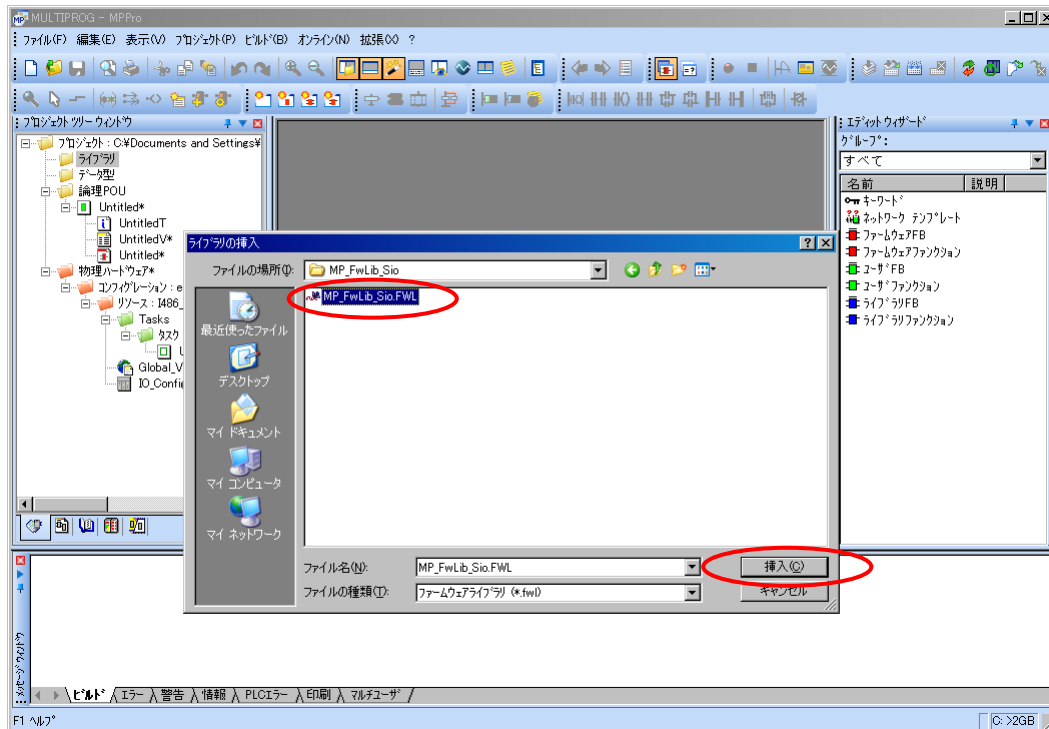


図 2-2-4. ライブラリ挿入画面③

- 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。

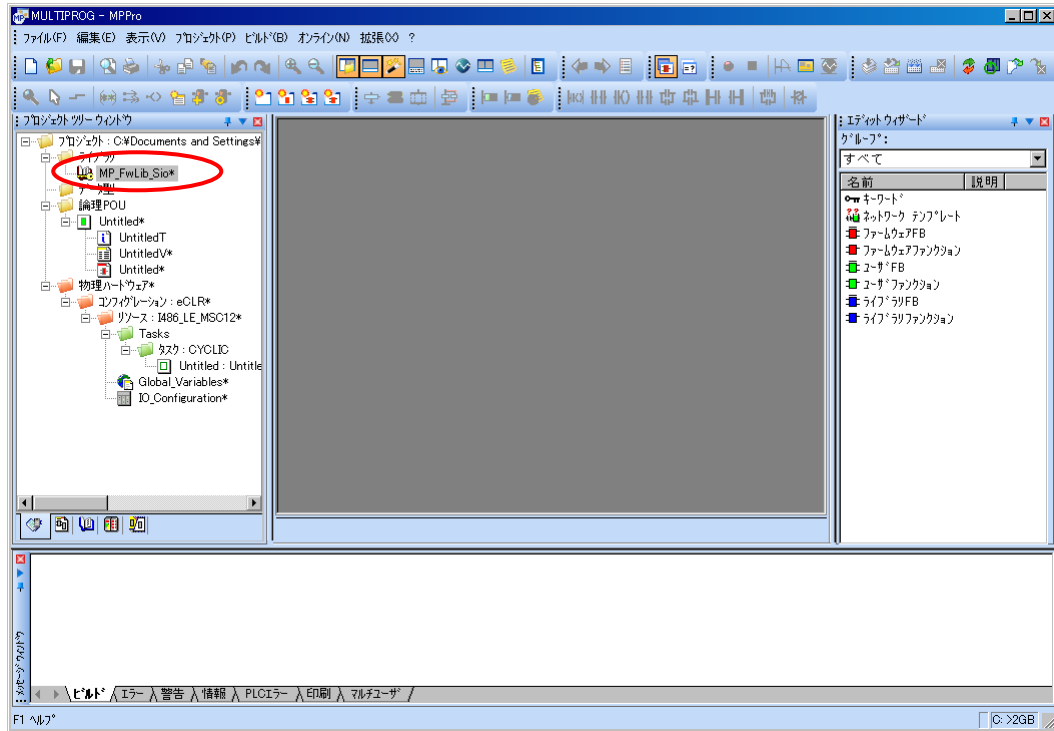


図 2-2-5. ライブラリ挿入画面④

## 2-3 ファンクションブロックリファレンス

本項では MULTIPROG 用として用意した標準 SIO の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コマンドを実行するための入力として「Act」、コマンドの応答結果を知らせるための出力として「Reply」「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

本パラメータが入力にある場合、立ち上りエッジにより処理を実行します。以降は処理完了待ち状態になります。

Done 出力が TRUE になる前に本パラメータを FALSE にする事で完了待ちを停止し、全ての出力を初期化しますが、命令のキャンセル処理ではありません。(送受信処理を途中で止めた場合、途中までは送受信されず。)

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。

### Enable

本パラメータが入力にある場合、TRUE にする事でその処理はサイクルスキャン毎に実行されます。

### 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

データ送受信 (ComSend、ComRecv) ファンクションブロックには、送受信用の電文格納用バッファとして ANY 型の Data という入出力が用意されています。

Data 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;
```

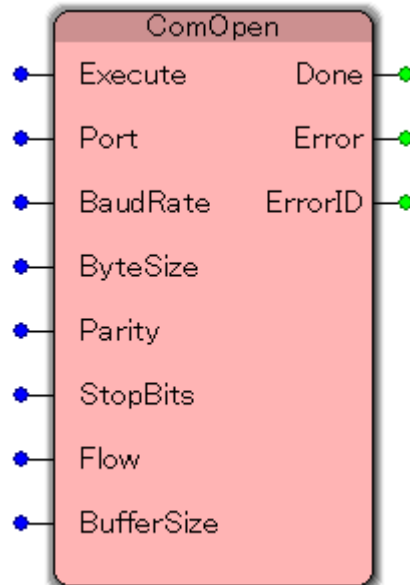
END\_TYPE

ここで BUFFER 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。

ComOpen FB

**機能** 通信設定と接続を行います。

**書式**



**入力**

BOOL	Execute	:	実行フラグ (立ち上りエッジにより実行)
INT	Port	:	COM ポート番号 [1 ~ 4]
UDINT	BaudRate	:	通信速度
INT	ByteSize	:	データビット長
INT	Parity	:	パリティビット
INT	StopBits	:	ストップビット長
BOOL	Flow	:	XON/XOFF フローコントロール有無
UDINT	BufferSize	:	内部バッファサイズ (送受信共通)

**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)
DINT	ErrorID	:	エラーコード (エラーコード一覧を参照)

**説明** シリアルポートの通信設定と接続を行います。  
 本FBの正常完了後、シリアルポート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シリアル通信を行う前に、必ずコールする必要があります。

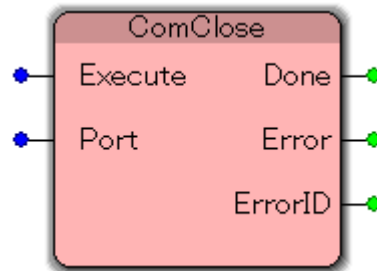
シリアル通信設定

- BaudRate** : 通信速度  
 [110, 300, 600, 1200, 2400, 4800, 9600]  
 [14400, 19200, 38400, 56000, 57600]  
 [115200, 128000, 256000] 単位 : 【bps】
- ByteSize** : データビット長  
 [5, 6, 7, 8] 単位 : 【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    |
| 3       | MARK : マークパリティ   |
| 4       | SPACE : スペースパリティ |
- StopBits** : 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   |
- Flow** : XON/XOFF フローコントロール有無  
 [ 0 : 無し]  
 [ 1 : 有り]
- BufferSize** : 内部バッファサイズ (送受信共通)  
 [ 1 ~ 65535 ] 単位 : 【バイト】

## ComClose FB

**機能** 通信切断を行います。

**書式**



**入力**

BOOL	Execute	:	実行フラグ (立ち上りエッジにより実行)
INT	Port	:	COM ポート番号 [ 1 ~ 4 ]

**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)
DINT	ErrorID	:	エラーコード (エラーコード一覧を参照)

**説明**

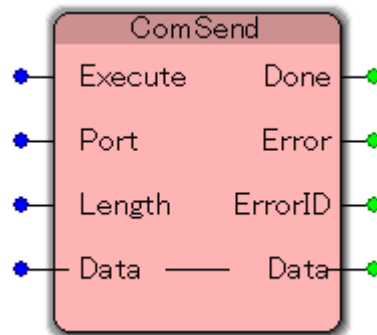
シリアル通信を切断します。  
 本 FB をコール後は、ComOpen 以外の FB の実行は不可になります。  
 本ライブラリの使用を終了する場合は、必ずコールする必要があります。



ComSend FB

**機能** 電文の送信を行います。

**書式**



**入力**

BOOL	Execute	:	実行フラグ (立ち上りエッジにより実行)
INT	Port	:	COM ポート番号 [1~4]
UDINT	Length	:	送信データ長 [1~内部バッファサイズ] 又は [1~送信データバッファサイズ]

**出力**

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

**入出力**

ANY	Data	:	送信データバッファ
-----	------	---	-----------

**説明**

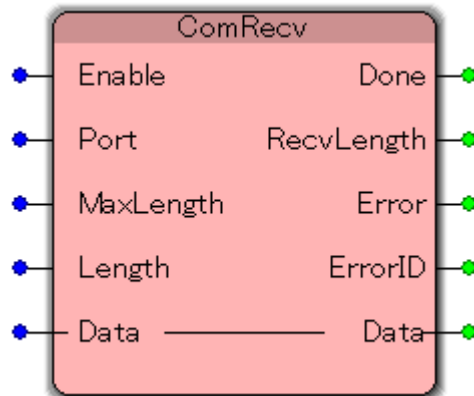
Data にセットされた電文の送信を行います。  
 送信が完了すると Done=TRUE になります。  
 送信は、Execute=TRUE としてから 1msec 以内に開始されます。  
 エラーコードに「送信リングバッファフル」が出力された場合は、送信バッファに対して ComPurge を実行してください。

ComRecv FB

**機能**

電文の受信を行います。

**書式**



**入力**

BOOL	Enable	:	継続実行フラグ (ONの間、実行)
INT	Port	:	COMポート番号 [1~4]
UDINT	MaxLength	:	受信データバッファサイズ
UDINT	Length	:	受信待ちデータ長 [1~内部バッファサイズ] 又は [1~受信データバッファサイズ]

**出力**

BOOL	Done	:	正常完了フラグ (0:応答なし, 1:受信完了)
UDINT	RecvLength	:	受信済みデータ長
BOOL	Error	:	エラーフラグ (0:エラーなし, 1:エラーあり)
DINT	ErrorID	:	エラーコード (エラーコード一覧を参照)

**入出力**

ANY	Data	:	受信データバッファ
-----	------	---	-----------

**説明**

受信した電文を Data にセットします。  
 Length 分のデータを受信完了すると、Done=TRUE になります。  
 Done=TRUE 以降、Data は変化しません。(受信電文は内部バッファに保持されます。)  
 Enable の立ち上りエッジにより、Data にセットされたバッファはクリアされ、バッファ先頭から受信データがセットされます。  
 エラーコードに「受信リングバッファフル」が出力された場合は、受信バッファに対して ComPurge を実行してください。  
 \*) ComPurge 無しでも本 FB を実行できますが、バッファ内のデータ順序は保障されません。  
 \*) スタートコード・エンドコードを設定している場合、Length の扱いや Done の条件が変わります。次ページを参照してください。

スタートコード・エンドコードを設定している場合、Length の扱いや Done の条件、Data バッファの内容は、表 2-3-1 のようになります。

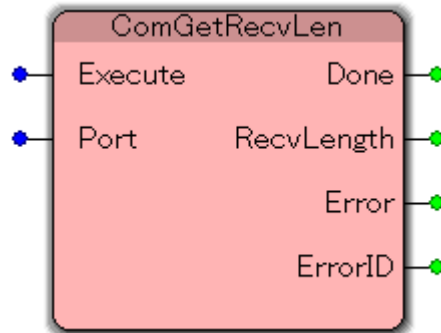
表 2-3-1. スタート・エンドコード設定時の ComRecv パラメータ

		スタート(S)・エンド(E)コード設定		
		Sのみ	Eのみ	S/E
入出力:	Data	S + 電文	電文 + E	S + 電文 + E
入力:	Enable			
	Port			
	MaxLength			
	Length	S含む	無効(1サイクルの読出サイズ)	
出力:	Done	指定電文長	エンドコード	エンドコード
	RecvLength	S + 電文	電文 + E	S + 電文 + E
	Error			
	ErrorID			

## ComGetRecvLen FB

**機能** 受信済み電文長を取得します。

**書式**



**入力**

BOOL	Execute	:	実行フラグ (立ち上りエッジにより実行)
INT	Port	:	COMポート番号[1~4]

**出力**

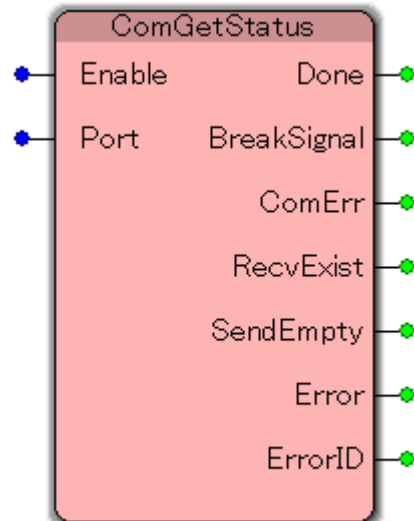
BOOL	Done	:	正常完了フラグ(0:応答なし, 1:受信完了)
UDINT	RecvLength	:	受信済みデータ長
BOOL	Error	:	エラーフラグ(0:エラーなし, 1:エラーあり)
DINT	ErrorID	:	エラーコード(エラーコード一覧を参照)

**説明** 内部バッファに受信済みの電文長を取得します。  
何度も電文長を取得する場合は、Execute の ON/OFF が必要になります。

ComGetStatus FB

**機能** 指定ポートの通信状態を取得します。

**書式**



**入力**

BOOL	Enable	:	継続実行フラグ (ONの間、実行)
INT	Port	:	COMポート番号[1~4]

**出力**

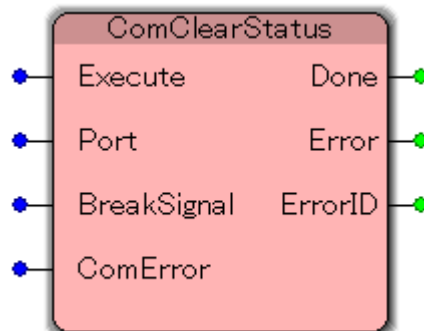
BOOL	Done	:	正常完了フラグ (0:応答なし, 1:受信完了)
BOOL	BreakSignal	:	ブレーク信号受信フラグ
BOOL	ComErr	:	回線状態エラーフラグ (フレーミング、オーバーラン、受信パリティ)
BOOL	RecvExist	:	受信データ有りフラグ
BOOL	SendEmpty	:	送信バッファエンプティーフラグ
BOOL	Error	:	エラーフラグ (0:エラーなし, 1:エラーあり)
DINT	ErrorID	:	エラーコード (エラーコード一覧を参照)

**説明** 指定ポートの通信状態を取得します。  
 回線状態エラーについては、個々の識別は出来ません。  
 ブレーク信号受信、回線状態エラーが発生した場合、再度の監視には ComClearStatus を実行する必要があります。

ComClearStatus FB

**機能** 通信ステータス情報をクリアします。

**書式**



**入力**

BOOL	Execute	:	実行フラグ (立ち上りエッジにより実行)
INT	Port	:	COM ポート番号 [1 ~ 4]
BOOL	BreakSignal	:	ブレーク受信ステータスクリアフラグ
BOOL	ComError	:	エラーステータスクリアフラグ

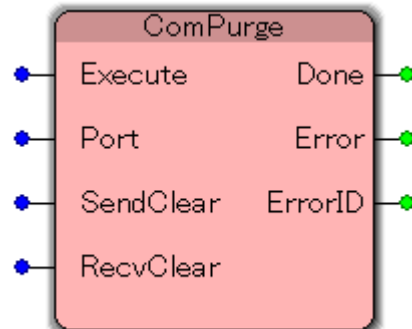
**出力**

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

**説明** 通信ステータス情報の内、ブレーク信号受信、回線状態エラーのクリアを行います。本FBを実行する事で、ComGetStatusで再度の監視が可能になります。

## ComPurge FB

**機能** 送信バッファ、受信バッファのクリアを行います。

**書式**

**入力**

BOOL	Execute	:	実行フラグ (立ち上りエッジにより実行)
INT	Port	:	COM ポート番号 [1 ~ 4]
BOOL	SendClear	:	送信バッファクリアフラグ (0:無し, 1:クリア)
BOOL	RecvClear	:	受信バッファクリアフラグ (0:無し, 1:クリア)

**出力**

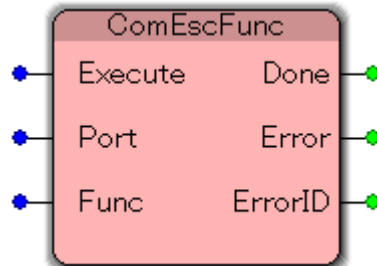
BOOL	Done	:	正常完了フラグ (0:応答なし, 1:受信完了)
BOOL	Error	:	エラーフラグ (0:エラーなし, 1:エラーあり)
DINT	ErrorID	:	エラーコード (エラーコード一覧を参照)

**説明** SendClear, RecvClear で指定されたバッファのクリアを行います。  
 クリアは、ライブラリ内部バッファ、ハードウェアバッファのそれぞれに対して実行します。  
 「送信リングバッファフル」エラーや「受信リングバッファフル」エラー、ComGetStatus の ComErr 出力が ON になった場合には、本 FB を実行してください。

### ComEscFunc FB

**機能** 通信拡張機能を実行します。

**書式**



**入力**

BOOL	Execute	:	実行フラグ (立ち上りエッジにより実行)
INT	Port	:	COM ポート番号 [ 1 ~ 4 ]
INT	Func	:	拡張機能番号

**出力**

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

**説明** 通信の拡張機能を実行します。  
 現在拡張機能として用意されているものは DTR, RTS 信号の操作になります。

#### 拡張機能番号

Func : 拡張機能番号

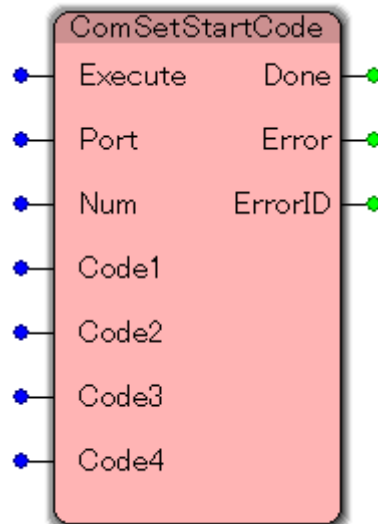
拡張機能番号	内容
1	DTR (データ端末準備完了) 信号を消去します。
2	RTS (送信要求) 信号を消去します。
3	DTR (データ端末準備完了) 信号を送信します。
4	RTS (送信要求) 信号を送信します。



ComSetStartCode FB

**機能** スタートコードを設定します。

**書式**



**入力**

BOOL	Execute	:	実行フラグ (立ち上りエッジにより実行)
INT	Port	:	COM ポート番号 [1 ~ 4]
BYTE	Num	:	スタートコード長
BYTE	Code1	:	1 バイト目スタートコード
BYTE	Code2	:	2 バイト目スタートコード
BYTE	Code3	:	3 バイト目スタートコード
BYTE	Code4	:	4 バイト目スタートコード

**出力**

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

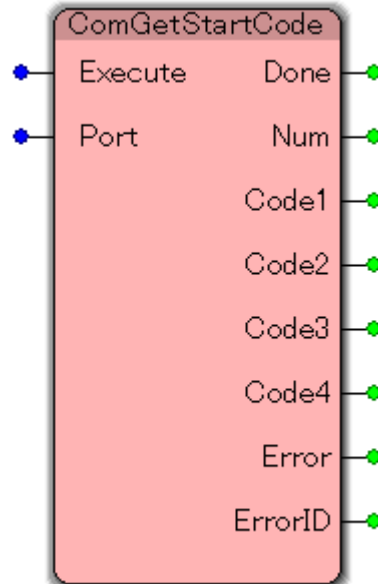
**説明** 通信電文の開始を示すスタートコードを設定します。  
 本 FB によりスタートコードの設定を行うと、スタートコード受信までの電文は読み捨てられます。

### ComGetStartCode FB

**機能**

設定済みのスタートコードを取得します。

**書式**



**入力**

BOOL Execute : 実行フラグ (立ち上りエッジにより実行)  
 INT Port : COMポート番号[1~4]

**出力**

BOOL Done : 正常完了フラグ(0:応答なし, 1:受信完了)  
 BYTE Num : スタートコード長  
 BYTE Code1 : 1バイト目スタートコード  
 BYTE Code2 : 2バイト目スタートコード  
 BYTE Code3 : 3バイト目スタートコード  
 BYTE Code4 : 4バイト目スタートコード  
 BOOL Error : エラーフラグ(0:エラーなし, 1:エラーあり)  
 DINT ErrorID : エラーコード(エラーコード一覧を参照)

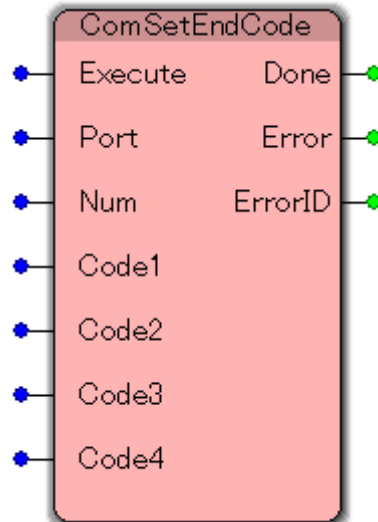
**説明**

設定済みのスタートコードを取得します。

ComSetEndCode FB

**機能** エンドコードを設定します。

**書式**



**入力**

BOOL	Execute	:	実行フラグ (立ち上りエッジにより実行)
INT	Port	:	COM ポート番号 [1 ~ 4]
BYTE	Num	:	エンドコード長
BYTE	Code1	:	1 バイト目エンドコード
BYTE	Code2	:	2 バイト目エンドコード
BYTE	Code3	:	3 バイト目エンドコード
BYTE	Code4	:	4 バイト目エンドコード

**出力**

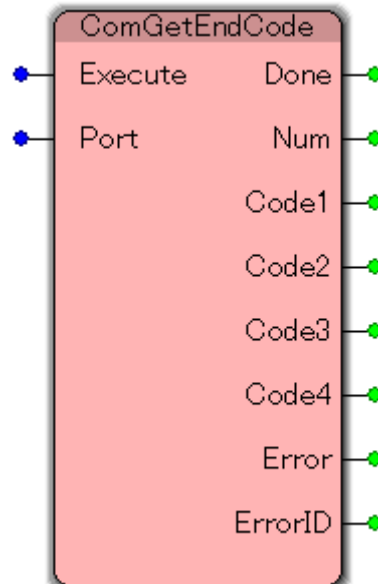
BOOL	Done	:	正常完了フラグ (0: 応答なし, 1: 受信完了)
BOOL	Error	:	エラーフラグ (0: エラーなし, 1: エラーあり)
DINT	ErrorID	:	エラーコード (エラーコード一覧を参照)

**説明** 通信電文の終了を示すエンドコードを設定します。  
 本 FB によりエンドコードの設定を行うと、エンドコード受信まで受信処理は完了しません。  
 (ComRecv の Done が ON になる条件になります。)

### ComGetEndCode FB

**機能** 設定済みのエンドコードを取得します。

**書式**



**入力**

BOOL	Execute	:	実行フラグ (立ち上りエッジにより実行)
INT	Port	:	COM ポート番号 [ 1 ~ 4 ]

**出力**

BOOL	Done	:	正常完了フラグ (0: 応答なし, 1: 受信完了)
BYTE	Num	:	エンドコード長
BYTE	Code1	:	1 バイト目エンドコード
BYTE	Code2	:	2 バイト目エンドコード
BYTE	Code3	:	3 バイト目エンドコード
BYTE	Code4	:	4 バイト目エンドコード
BOOL	Error	:	エラーフラグ (0: エラーなし, 1: エラーあり)
DINT	ErrorID	:	エラーコード (エラーコード一覧を参照)

**説明** 設定済みのエンドコードを取得します。

## 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	出力 FB
ERR_CSIO_PORT	-1	ポート番号 仕様範囲外	すべての FB
ERR_CSIO_ALRADYOPEN	-2	既に Open 済み	ComOpen
ERR_CSIO_NOTOPEN	-3	Open 未完	ComOpen を除くすべての FB
ERR_CSIO_PARAMETER	-4	引数異常	ComPurge、ComEscFunc ComSetStartCode、ComSetEndCode
ERR_CSIO_SENDBUF	-5	送信リングバッファフル	ComSend
ERR_CSIO_RECVBUF	-6	受信リングバッファフル	ComRecv
ERR_CSIO_PARAM_BAUD	-7	ボーレート 仕様範囲外	ComOpen
ERR_CSIO_PARAM_BYTE	-8	データビット 仕様範囲外	ComOpen
ERR_CSIO_PARAM_PARITY	-9	パリティビット 仕様範囲外	ComOpen
ERR_CSIO_PARAM_STOP	-10	ストップビット 仕様範囲外	ComOpen
ERR_CSIO_PARAM_LENGTH	-11	Length 異常	ComSend
		Length, MaxLength 異常	ComRecv

## 内部エラーコード一覧

表 2-4-2. 内部エラーコード一覧

定義名	コード	内容	出力 FB
ERR_CSIO_COMOPEN	-100	OpenComm 失敗	ComOpen
ERR_CSIO_PURGE	-101	PurgeComm 失敗	ComPurge
ERR_CSIO_ESCFUNC	-102	EscapeCommFunction 失敗	ComEscFunc
ERR_CSIO_CLRBREAK	-103	ClearBreak 失敗	ComClearStatus
ERR_CSIO_CLRERROR	-104	ClearError 失敗	ComClearStatus

## 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。

77KW10022E  
77KW10022A

2021年 8月 第5版  
2013年 10月 初版

 株式会社アルゴシステム

本社  
〒587-0021 大阪府堺市美原区小平尾656番地

TEL(072)362-5067  
FAX(072)362-4856

ホームページ <http://www.algosystem.co.jp>