




# User Manual

## Anybus<sup>®</sup> Communicator<sup>™</sup> for EtherNet/IP

Doc.Id. JCM-1201-003  
Rev. 2.04

---

### HMS Industrial Networks AB

   
Germany +49 - 721 - 96472 - 0  
Japan +81 - 45 - 478 -5340  
Sweden +46 - 35 - 17 29 20  
U.S.A. +1 - 312 - 829 - 0601  
France +33 - 3 89 32 76 76  
Italy +39 - 347 - 00894 - 70  
China +86 - 10 - 8532 - 3183

   
ge-sales@hms-networks.com  
jp-sales@hms-networks.com  
sales@hms-networks.com  
us-sales@hms-networks.com  
fr-sales@hms-networks.com  
it-sales@hms-networks.com  
cn-sales@hms-networks.com





# 目次

序章	このマニュアルについて	
	このドキュメントの使い方 .....	P-1
	重要なユーザ情報 .....	P-1
	関連マニュアル .....	P-2
	マニュアル更新履歴 .....	P-2
	慣例と用語集 .....	P-3
	用語集 .....	P-3
	サポート .....	P-4
第1章	EtherNet/IP 向け Anybus Communicator について	
	外観 .....	1-2
	LED ステータス .....	1-3
	コンフィグレーション・スイッチ .....	1-4
	ハードウェアのインストール .....	1-5
	ソフトウェアのインストール .....	1-6
	ABC コンフィグツール .....	1-6
第2章	基本操作	
	概論 .....	2-1
	データ交換モデル1 .....	2-2
	メモリマップ .....	2-2
	データ交換例 .....	2-3
	サブネットワーク・プロトコル .....	2-4
	プロトコル・モード .....	2-4
	プロトコル構築ブロック .....	2-4
	マスター・モード .....	2-5
	ジェネリック・データ・モード .....	2-5
	EtherNet/IP .....	2-6
	概論 .....	2-6
	データ・タイプ .....	2-6
	メモリ・レイアウト .....	2-6
	Modbus/TCP .....	2-7
	概論 .....	2-7
	アドレッシング・モード .....	2-7
	サポートされている例外コード .....	2-7
	Modbus アドレッシング・モード .....	2-8
	Anybus アドレッシング・モード .....	2-9
第3章	ファイル・システム	
	概論 .....	3-1
	概要 .....	3-2
	システム・ファイル .....	3-2

第 4 章	基本的なネットワーク・コンフィグレーション	
	概論.....	4-1
	Ethernet コンフィグレーション・ファイル ('ethcfg.cfg') .....	4-2
	概論.....	4-2
	IP アクセス制御 .....	4-3
	Anybus IPconfig (HICP) .....	4-4
第 5 章	FTP サーバ	
	概論.....	5-1
第 6 章	Telnet サーバ	
	概論.....	6-1
	一般的なコマンド .....	6-2
	診断コマンド .....	6-3
	ファイル・システム操作.....	6-3
第 7 章	Web サーバ	
	概論.....	7-1
	許可.....	7-2
	内容の種類.....	7-3
第 8 章	Server Side Include (SSI)	
	機能.....	8-2
	SSI 出力の変更.....	8-9
	SSI 出力ストリング・ファイル.....	8-9
	一時的な SSI 出力変更.....	8-10
第 9 章	Email クライアント	
	概論.....	9-1
	Email 定義 .....	9-2
第 10 章	ABC コンフィグツールのナビゲーション	
	メイン・ウィンドウ.....	10-1
	プルダウン・メニュー.....	10-2
	ツールバー・アイコン.....	10-5
第 11 章	基本設定	
	フィールドバス設定.....	11-1
	ABC パラメータ .....	11-3
	サブネットワーク・パラメータ .....	11-4
第 12 章	ノード	
	概論.....	12-1
	ノードの追加と管理.....	12-1
	ノード・パラメータ .....	12-1

第 13 章	トランザクション	
	概論.....	13-1
	トランザクションの追加と管理.....	13-1
	トランザクション・パラメータ (マスター・モード) .....	13-2
	パラメータ (クエリとブロードキャスト) .....	13-2
	パラメータ (レスポンス) .....	13-3
	トランザクション・パラメータ (ジェネリック・データ・モード) .....	13-3
	プロデュース・トランザクション (Produce- Transactions) .....	13-3
	コンスーム・トランザクション (Consume- Transactions).....	13-4
	トランザクション・エディタ .....	13-5
第 14 章	フレーム・オブジェクト	
	概論.....	14-1
	フレーム・オブジェクトの追加と編集 .....	14-1
	通常オブジェクト (バイト、ワード、ダブルワード) .....	14-2
	制限オブジェクト (バイト、ワード、ダブルワード) .....	14-3
	データ・オブジェクト .....	14-4
	変数データ・オブジェクト .....	14-4
	チェックサム・オブジェクト .....	14-6
第 15 章	コマンド	
	概論.....	15-1
	コマンドの追加と管理 .....	15-1
	プルダウン・メニュー .....	15-2
	Toolbar Icons .....	15-2
	コマンド・エディタ .....	15-3
	概論.....	15-3
	基本ナビゲーション.....	15-3
	プルダウン・メニュー .....	15-4
	コマンドの編集.....	15-5
	例：マスター・モードでの Modbus-RTU コマンドの指定 .....	15-6
第 16 章	サブネットワーク・モニタ	
第 17 章	ノード・モニタ	
	概論.....	17-1
	ノード・モニタのナビゲーション .....	17-2
	プルダウン・メニュー .....	17-3
	ツールバー・アイコン .....	17-4
第 18 章	データ・ロガー	
	概論.....	18-1
	操作.....	18-1
	コンフィグレーション .....	18-2

第 19 章	コンフィグレーション・ウィザード	
	概論.....	19-1
	ウィザード・プロファイルの選択.....	19-1
	ウィザード - Modbus RTU マスター.....	19-2
第 20 章	制御 / ステータス・レジスタ	
	概論.....	20-1
	ハンドシェーク手順.....	20-1
	データの一貫性.....	20-2
	ステータス・レジスタ構成（ゲートウェイから制御システムへ）.....	20-3
	概論.....	20-3
	マスター・モードでのステータス・コード.....	20-3
	ジェネリック・データ・モードでのステータス・コード.....	20-4
	制御レジスタ構成（制御システムからゲートウェイへ）.....	20-5
	概論.....	20-5
	マスター・モードでの制御コード.....	20-5
	ジェネリック・データ・モードでの制御コード.....	20-5
第 21 章	CIP オブジェクトの実装	
	概論.....	21-1
	識別オブジェクト、クラス 01h.....	21-2
	概論.....	21-2
	クラス属性.....	21-2
	インスタンス属性.....	21-2
	メッセージ・ルータ、クラス 02h.....	21-3
	概論.....	21-3
	クラス属性.....	21-3
	インスタンス属性.....	21-3
	アセンブリ・オブジェクト、クラス 04h.....	21-4
	概論.....	21-4
	クラス属性.....	21-4
	インスタンス属性 - インスタンス / 接続ポイント 64b.....	21-4
	インスタンス属性 - Instance / Connection Point 96b.....	21-4
	診断オブジェクト、クラス AAh.....	21-5
	概論.....	21-5
	クラス属性.....	21-5
	インスタンス属性, Instance 01h.....	21-5
	パラメータ・データ入力マッピング・オブジェクト、クラス B0h.....	21-6
	概論.....	21-6
	クラス属性.....	21-6
	インスタンス属性、インスタンス 01h.....	21-6
	パラメータ・データ出力マッピング・オブジェクト、クラス B1h.....	21-7
	概論.....	21-7
	クラス属性.....	21-7
	インスタンス属性、インスタンス 01h.....	21-7
	ポート・オブジェクト、クラス F4h.....	21-8
	概論.....	21-8
	クラス属性.....	21-8
	インスタンス属性、インスタンス 02h.....	21-8

TCP/IP インターフェース・オブジェクト、クラス F5h.....	21-9
概論.....	21-9
クラス属性.....	21-9
インスタンス属性.....	21-9
Ethernet リンク・オブジェクト、クラス 6Fh.....	21-10
概論.....	21-10
クラス属性.....	21-10
インスタンス属性.....	21-11
<b>第 22 章</b>	<b>先進フィールドバス・コンフィグレーション</b>
概論.....	22-1
メールボックス・エディタ .....	22-1
<b>アペンディックス A</b>	<b>パラメータ・データの初期化（エクスプリシット・データ）</b>
概論.....	A-1
メールボックス・メッセージの追加.....	A-1
Ethernet/IP への入力パラメータ・データのマッピング .....	A-2
Ethernet/IP への出力パラメータ・データのマッピング .....	A-4
<b>アペンディックス B</b>	<b>コネクタピン割り付け</b>
Ethernet コネクタ .....	B-1
電源コネクタ .....	B-1
PC コネクタ .....	B-2
サブネットワーク・インターフェース .....	B-3
概論.....	B-3
バイアス・レジスタ (RS485 のみ) .....	B-3
終端 (RS485とRS422 のみ) .....	B-3
コネクタ ピン配置 (DB9F).....	B-4
代表的な接続 (RS485) .....	B-5
代表的な接続 (RS422 と 4-ワイヤ RS485).....	B-5
代表的な接続 (RS232) .....	B-5
<b>アペンディックス C</b>	<b>技術仕様</b>
メカニカル仕様.....	C-1
電気的特性.....	C-1
環境特性.....	C-1
法的規制への遵守 .....	C-2
<b>アペンディックス D</b>	<b>トラブルシューティング</b>
<b>アペンディックス E</b>	<b>ASCII テーブル</b>





## このマニュアルについて

## このドキュメントの使い方

このマニュアルには PC ベースのコンフィグレーション・ソフトウェアを含む Anybus Communicator の概論と技術的機能が含まれています。

このマニュアルの読者は PLC、ソフトウェア設計、そして通信システムにも詳しいことが前提とされています。また、Microsoft Windows オペレーティング・システムにも慣れていることが前提とされています。

## 重要なユーザ情報

このマニュアルのデータと図は強制するものではありません。我々 HMS Industrial Networks AB は製品開発の継続のポリシーにより製品を変更する権利があります。このマニュアルの情報は通知をすることなく変わる内容で HMS Industrial Networks AB による委託であるとみなされません。

HMS Industrial Networks AB はこのマニュアルに現れるあらゆるエラーに対して責任を負いません。

この製品には多くのアプリケーションがあります。このデバイスの使用についての責任は、アプリケーションが適用される法律、規則、コードと規格を含む全ての動作と安全性の要求に適合する全ての必要な工程が行われたことが保証しなければなりません。

Anybus® は HMS Industrial Networks AB の登録商標です。全ての登録商標は所有者の資産です。

このマニュアルの例と図は実例を挙げるのが目的です。特殊な実装に関連した多くの変数や要件があるため、HMS Industrial Networks はこれらの例と図に基づいた実際の使用に対して責任または義務を保証することができません。

<b>警告：</b>	これはクラス A 製品です。使用される環境によって製品は電波干渉の原因になる可能性があります、その場合にはお客様にて十分な検証が必要になる場合があります。
------------	---

<b>ESD Note:</b>	この製品は ESD（放電）に敏感な部分が含まれているため、ESD 対策が十分でない場合には破損する可能性があります。製品を直接手で扱うときは静電気対策が必要です。これらを行わないと製品を破損させる可能性があります。
------------------	---

[illegible]

変更内容	Page(s)
マイナー修正と更新	-
ステータス LED 表示更新	1-3
“IO Size In” と “IO Size Out” パラメータの使用の明確化	11-1
“On data change” パラメータ動作の更新	13-2 - 13-4

[illegible]

## 慣例と用語集

本マニュアルでは下記の慣例を使用しています。

- 番号をつけたリストが連番で提供されます。
- 括弧をつけたリストは情報を提供していますが手順ではありません。
- ここでいう「ユーザ」とは、ネットワークで Anybus Communicator の実装に携わる人、または人々を意味しています。
- ‘ABC’ という用語は Anybus Communicator を意味します。
- 16 進値は 0xNNNN というフォーマットで書かれています。NNNN は 16 進法の値です。
- 10 進値は NNNN と表記されています。NNNN は 10 進法の値です。
- すべての通信システムにおいて「入力」「出力」という用語はあいまいです。その意味は最終的に参照される箇所次第だからです。本ドキュメントの慣例では「入力」「出力」はいつもマスター / スキャナを参照しています。

## 用語集

用語	説明
ABC	Anybus <sup>®</sup> Communicator <sup>™</sup>
ブロードキャスト	全てのノードに送信されるトランザクションを処理するコンフィグレーション内のプロトコル固有のノード
EIP	EtherNet/IP
コマンド	事前定義されたトランザクション
コンフィグレーション	トランザクションによってコンフィグレーションされたサブネットワーク上のノードのリスト
フィールドバス	コミュニケータに接続された上位レベル・ネットワーク
フィールドバス・コントロール・システム	フィールドバス・マスター
フレーム・オブジェクト	トランザクションの異なる部分を記述するために使用されるローレベル・エンティティ
モニタ	ABC とネットワーク接続
ノード	サブネットワーク上のノードとの通信を定義するコンフィグレーション内のデバイス
サブネットワーク	理論的にはフィールドバスの補助レベル上に位置するネットワーク。ゲートウェイのような役割を果たす ABC
トランザクション	サブネットワーク・コンフィグレーションで使用されるジェネリック・ビルディング・ブロックでサブネットに送受信されるデータを定義。
ユーザ	Anybus Communicator を実装する責任を持つ人 / 人々
上位レベル・ネットワーク	この場合 EtherNet (EtherNet/IP と Modbus/TCP を含む)
ネットワーク	
フィールドバス	

## サポート

技術サポートに関してオンライン FAQ ([www.anybus.com](http://www.anybus.com)) か、最寄りのサポート・センターまでご連絡ください。

### HMS Sweden (Head Office)

E-mail: [support@hms-networks.com](mailto:support@hms-networks.com)  
Phone: +46 (0) 35 - 17 29 20  
Fax: +46 (0) 35 - 17 29 09  
Online: [www.anybus.com](http://www.anybus.com)

### HMS North America

E-mail: [us-support@hms-networks.com](mailto:us-support@hms-networks.com)  
Phone: +1-312-829-0601  
Toll Free: +1-888-8-Anybus  
Fax: +1-312-738-5873  
Online: [www.anybus.com](http://www.anybus.com)

### HMS Germany

E-mail: [ge-support@hms-networks.com](mailto:ge-support@hms-networks.com)  
Phone: +49-721-96472-0  
Fax: +49-721-964-7210  
Online: [www.anybus.com](http://www.anybus.com)

### HMS Japan

E-mail: [jp-support@hms-networks.com](mailto:jp-support@hms-networks.com)  
Phone: +81-45-478-5340  
Fax: +81-45-476-0315  
Online: [www.anybus.com](http://www.anybus.com)

### HMS China

E-mail: [cn-support@hms-networks.com](mailto:cn-support@hms-networks.com)  
Phone: +86 10 8532 3023  
Online: [www.anybus.com](http://www.anybus.com)

### HMS Italy

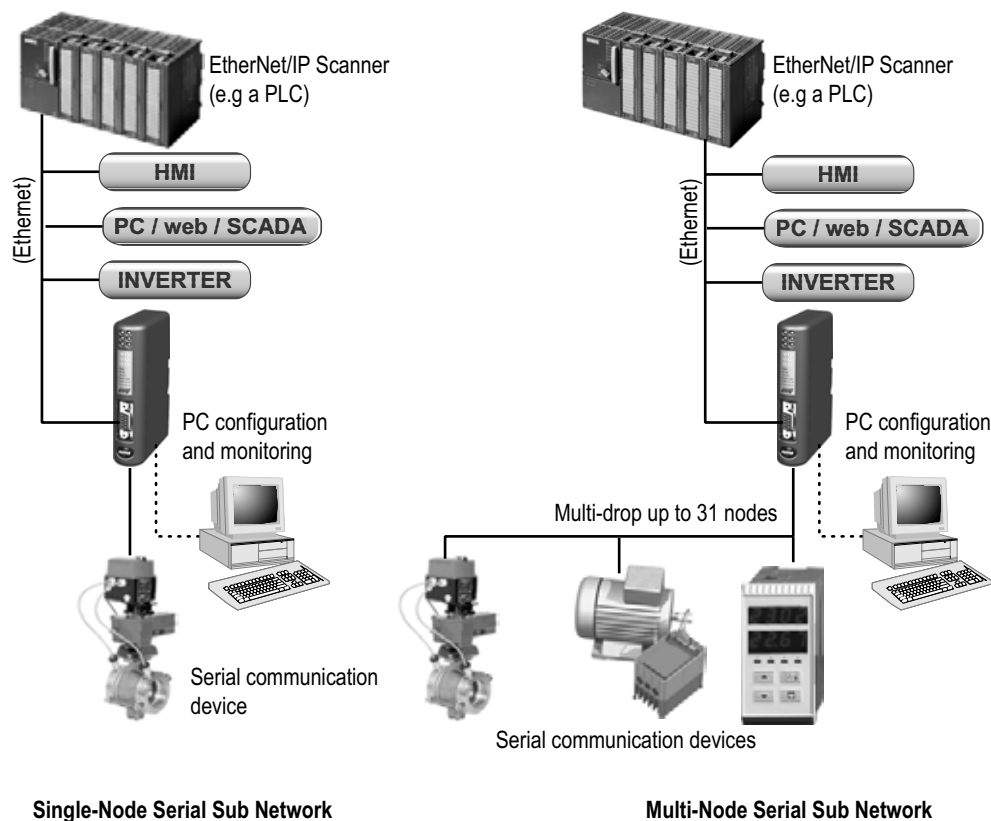
E-mail: [it-support@hms-networks.com](mailto:it-support@hms-networks.com)  
Phone: +39 039 59662 27  
Fax: +39 039 59662 31  
Online: [www.anybus.com](http://www.anybus.com)

### HMS France

E-mail: [fr-support@hms-networks.com](mailto:fr-support@hms-networks.com)  
Phone: +33 (0) 3 89 32 76 41  
Fax: +33 (0) 3 89 32 76 31  
Online: [www.anybus.com](http://www.anybus.com)

## EtherNet/IP 向け Anybus Communicator について

EtherNet/IP 向け Anybus Communicator、あるいは‘ABC’は、シリアル・アプリケーション・プロトコルと EtherNet/IP ネットワークの仮想的なゲートウェイとして機能します。産業機器との統合は新たにインストールをするときと同様に既存の機器と機能、制御、信頼性を失うことなく行うことができます。



### サブネットワーク

ゲートウェイは 31 ノードまでアドレスでき、物理的には次の標準をサポートします。

- RS-232
- RS-422
- RS-485

### Ethernet インターフェース

Ethernet 接続は特許取得、世界中の産業用オートメーション製品のメーカーに使用され証明された Anybus テクノロジーにより実現されます。

- Ethernet/IP グループ 2 と 3 サーバ
- Modbus/TCP スレーブ機能
- Server Side Include (SSI) 機能
- Web サーバ
- E メール・クライアント
- FTP & Telnet サーバ
- 10/100 M ビット / 秒、ツイスト・ペア

## 外観

結線とピン配置 B-1 “コネクタピン割り付け” を参照してください。

### A: EtherNet コネクタ

このコネクタは ABC をフィールドバスへ接続するために使用します。

- B-1 “Ethernet コネクタ” も参照してください。

### B: コンフィグレーション・スイッチ

- 1-4 “コンフィグレーション・スイッチ” も参照してください。

### C: LED ステータス

- 1-3 “LED ステータス” も参照してください。

### D: PC コネクタ

このコネクタはコンフィグレーションとモニタのために ABC を PC に接続するために使用します。

- B-2 “PC コネクタ” も参照してください。

### E: サブネットワーク・コネクタ

このコネクタは ABC とシリアル・サブネットワークを接続するために使用します。

- B-2 “PC コネクタ” も参照してください。

### F: 電源コネクタ

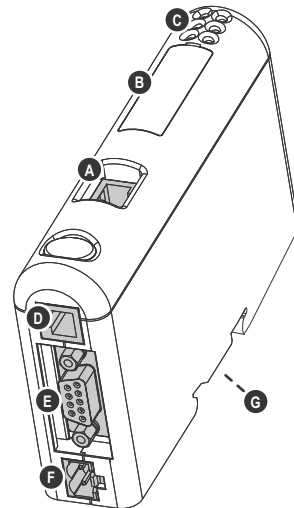
このコネクタは電源を ABC に供給するために使用します。

- B-1 “電源コネクタ”
- B-1 “技術仕様” も参照してください。

### G: DIN レール・コネクタ

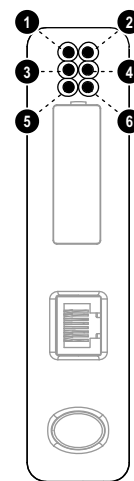
DIN レール・メカニズムは ABC と PE(保護アース)を接続するために使用されます。

- 1-5 “ハードウェアのインストール” も参照してください。



## LED ステータス

#	ステート	ステータス
1- モジュール・ステータス (EtherNet/IP 接続関連のみ)	オフ	電源なし
	グリーン	正常動作
	グリーン、点滅	コンフィグレーションされていない、またはスキャナが Run ステートでない
	レッド	メジャー・フォルト
	レッド、点滅	マイナー・フォルト
	レッド / グリーン 交互点滅	(セルフテスト)
2- ネットワーク・ステータス (EtherNet/IP 接続関連のみ)	オフ	IP アドレスなし (または電源なし)
	グリーン	EtherNet/IP 接続確立
	グリーン、点滅	確立された EtherNet/IP 接続なし
	レッド	重複 IP アドレス検出
	レッド、点滅	ひとつまたは複数の接続がタイムアウト
	レッド / グリーン 交互点滅	(セルフテスト)
3- リンク	オフ	リンクなし (または電源なし)
	グリーン	ethernet ネットワークへの接続
4- アクティビティ	オフ	ethernet アクティビティがない (または電源なし)
	グリーン	ethernet パケットの受信または送信
5- サブネット・ステータス <sup>a</sup>	オフ	電源なし
	グリーン、点滅	正常動作中だが一つ以上のトランザクション・エラーがある
	グリーン	動作中
	レッド	トランザクション・エラー / タイムアウトまたはサブネットが停止
6- デバイス・ステータス	オフ	電源なし
	レッド / グリーン 交互点滅	無効あるいはコンフィギュレーションのエラー
	グリーン	初期化中
	グリーン、点滅	動作中
	レッド、点滅	HMS サポートまでご連絡ください。



- a. この LED は全てのトランザクションが最低一度でもアクティブになったとき緑に変わります。これは “change of state” または “change of state on trigger” を使用したどんなトランザクションも含まれます。トランザクションでタイムアウトが発生した場合は、この LED は赤に変わります。

# コンフィグレーション・スイッチ

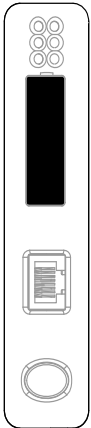
ゼロ以外の値が設定される場合、コンフィグレーション・スイッチで192.168.0.1 – 192.168.0.255 の範囲で IP アドレスの設定が必要です。ゼロに設定されるとこれらの設定はシステム・ファイル ‘ethcfg.cfg’ で設定されるか、ABC コンフィグツールで設定されます。

スイッチはスタートアップで読まれ、変更を有効にするためにリセットが必要です。

SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8	DHCP	Subnet	Gateway	IP
OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	'ethcfg.cfg' で設定			
OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON	OFF	255.255.255.0	192.168.0.255	192.168.0.1
OFF	OFF	OFF	OFF	OFF	OFF	ON	OFF	OFF	255.255.255.0	192.168.0.255	192.168.0.2
...	...	...	...	...	...	...	...	...	...	...	...
ON	ON	ON	ON	ON	ON	ON	OFF	OFF	255.255.255.0	192.168.0.255	192.168.0.254
ON	ON	ON	ON	ON	ON	ON	ON	無効な設定			

- 4-1 “ 基本的なネットワーク・コンフィグレーション ”
- 11-1 “ フィールドバス設定 ”

も参照してください。



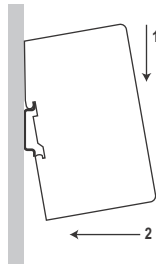


## ハードウェアのインストール

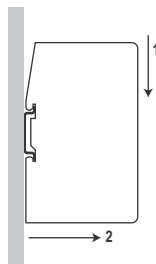
ゲートウェイを物理的にインストールするときは次のステップに従います。

1. ABC を DIN レールにパチッと音がするまで取り付ける (1-2 “DIN レール・コネクタ” を参照してください。)

DIN レール・メカニズムは次のように動作します。



ABC を取り付けるにはまず ABC を下のほうに押し (1)、DIN レール・メカニズムのスプリングに押し込み、今度はきっちりハマるように DIN レールにさからうように押します。(2)



ABC を取り外すには下方に押し (1) DIN レールから外れるように引き出します。(2)

2. ABC を etherNet ネットワークへ接続します。
3. ABC をシリアル・サブネットワークに接続します。
4. PC - ケーブルを介し ABC を PC の未使用の COM- ポートに接続します。
5. 基板上のスイッチで DeviceNet のボーレートと Mac-ID を設定します。
6. 電源ケーブルを接続し電源を供給します。  
(ABC コンフィグツール・ソフトウェアは自動的にシリアル・ポートを検索します。  
うまくいかなければマニュアルで “Port” メニューで正しいポートを選択します。)
7. PC で ABC コンフィグツールをスタートします。
8. ABC コンフィグレーションに従って EtherNet/IP 通信を設定します。

## ソフトウェアのインストール

### ABC コンフィグツール

#### システム要件

- Pentium 133 MHz 以上
- ハードドライブに 10 MB の空き容量
- 8 MB RAM
- 800x600 (16 ビット・カラー) 以上のスクリーン解像度
- Microsoft Windows™ NT4 / 2000 / XP
- Internet Explorer 4.01 SP1 以降

#### インストール

- **Anybus Communicator リソース CD**

CD を挿入しスクリーン上のインストラクションに従います。インストールが自動的に開始されなければ CD ドライブ・アイコンを右クリックし、Explore を選択します。‘setup.exe’ を実行し、スクリーンに表示される指示に従います。

- **Website から**

ダウンロードして自動解凍を実行します。exe-file は HMS の website からダウンロードできます。(www.anybus.com)

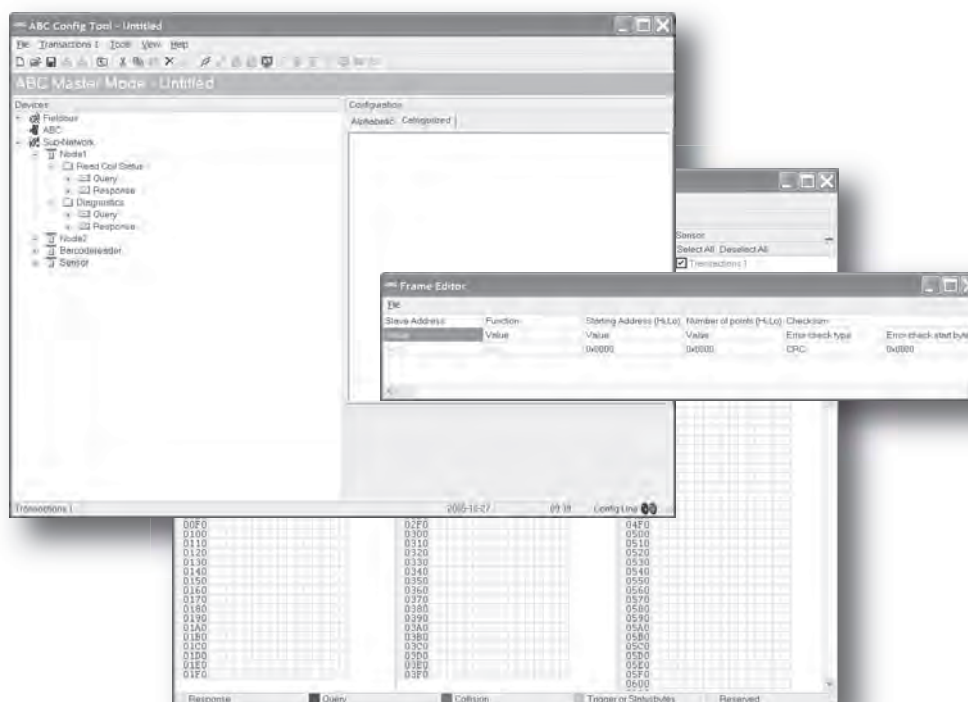
## 基本操作

### 概論

Anybus Communicator ゲートウェイはシリアル・サブネットワークと上位レベルのネットワークのデータ交換を行うようにデザインされています。似たような他のゲートウェイ機器とは異なりサブネットワーク向けの固定されたプロトコルを持たず、シリアル通信のほとんどのフォームを扱うようコンフィグレーションされています。

ABC は周期的にシリアル・テレグラムを発行することができます。ステートの変更、上位ネットワーク（フィールドバス・マスターあるいは PLC）の制御システムから発行されるトリガ・イベントをベースにしています。サブネットワーク通信をモニタすることができ、またデータが変更されたときに上位のネットワークに通知することができます。

基本的な Anybus Communicator パッケージは ABC コンフィグツール、Windows™ アプリケーションでゲートウェイにサブネットワーク・プロトコルの記述を提供するために使用されます。プログラミング不要で視覚的なプロトコル記述システムが異なるシリアル通信の一部を特定するために使用されます。



## データ交換モデル I

サブネットワークと上位ネットワーク上で交換されるデータは同じメモリ内にあります。

サブネットワークでデータを交換するために、上位ネットワークは単にABCコンフィグツールを使用して特定されたメモリロケーションにデータを読み書きします。そして同じメモリロケーションがサブネットワーク上で交換されます。

内部メモリ・バッファは機能に基づいて 3 つの領域に分けられます。

- **入力データ**

この領域は上位ネットワークにより読み込まれます。

(上位のネットワークでこのデータがどのように表されるかは本章で記述されます。)

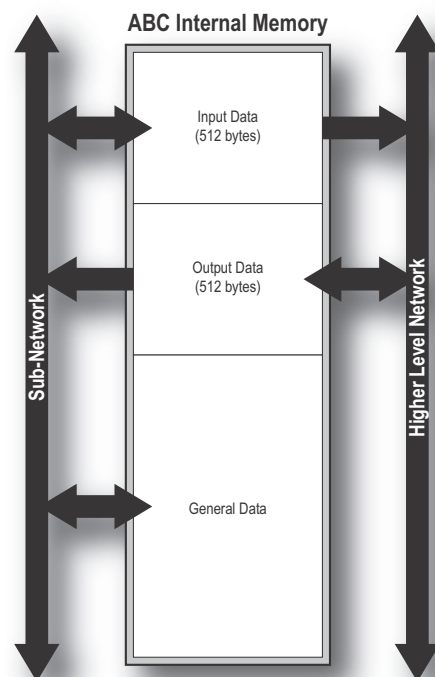
- **出力データ**

この領域は上位ネットワークによって書き込まれます。

(上位のネットワークでこのデータがどのように表されるかは本章で記述されます。)

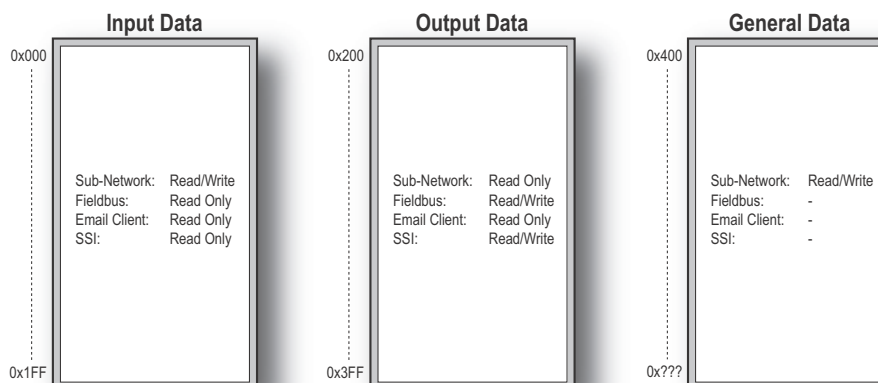
- **一般データ**

この領域は上位ネットワーク上で交換されず、サブネットワーク上のノード間の転送、あるいはデータの一般“scratch pad”に使用することができます。この領域の実サイズはサブネット上で交換されたデータ量により異なります。ABC は最大 1024 バイトの一般データを処理することができます。



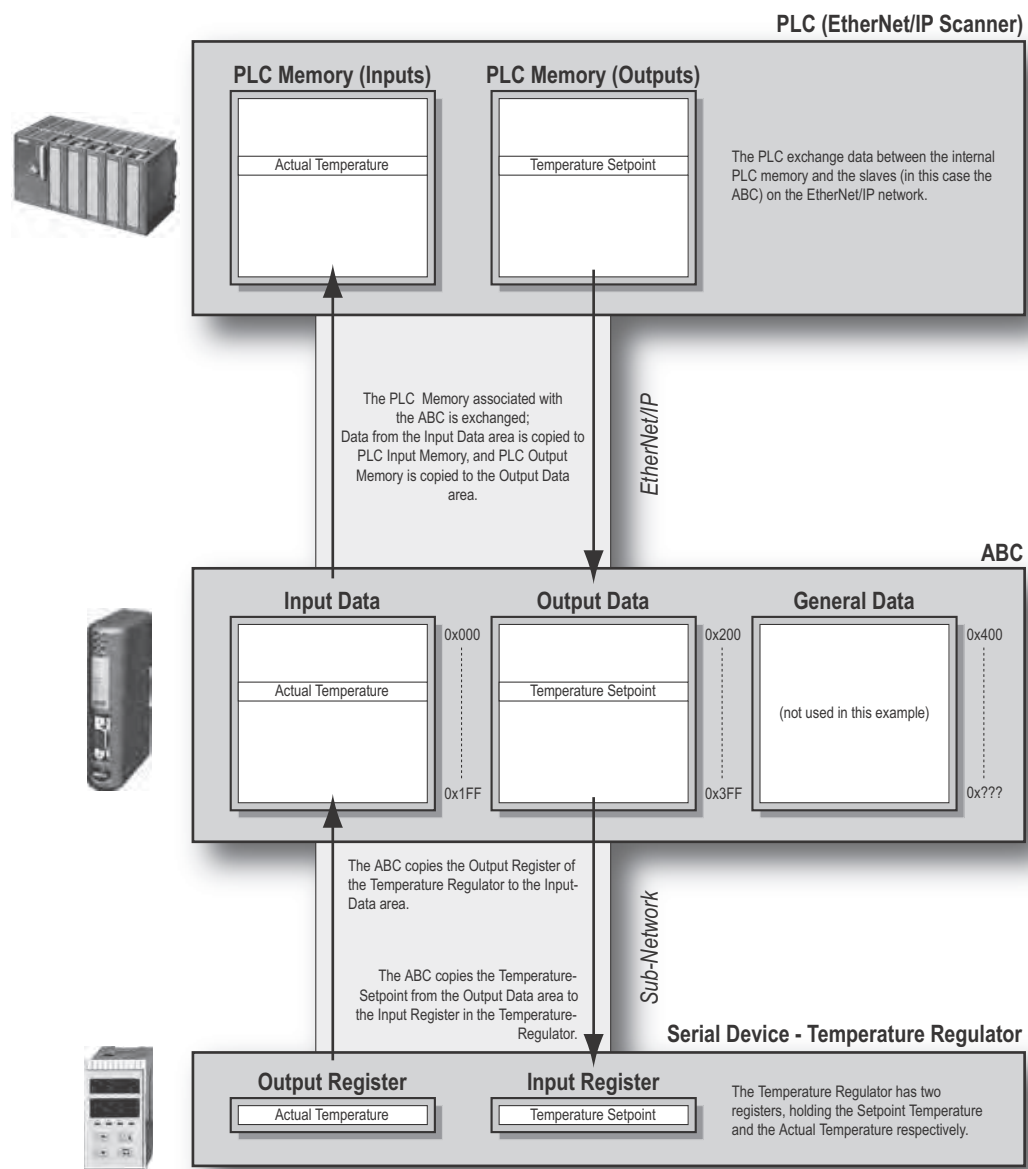
## メモリマップ

ABC コンフィグツールを使用してサブネットワークコンフィギュレーションを構築するとき、上記の異なる領域はメモリロケーション（アドレス）に次のように特定され、マッピングされます。



## データ交換例

次の例ではサブネットワーク上の温度調整器が上位ネットワーク上の PLC と ABC のメモリ・バッファを介して情報を交換しています。



## サブネットワーク・プロトコル

### プロトコル・モード

ABC には‘マスター・モード’と‘ジェネリック・データ・モード’と呼ばれる 2 つのサブネットワーク通信操作モード機能があります。

- マスター・モード

このモードでは ABC はサブネット上でマスターの役割をし、クエリ / レスポンス方式でシリアル通信が行われます。ネットワーク上のノードはまず ABC にアドレスされなければメッセージを発行することが許されません。

2-5 “マスター・モード” を参照してください。

- ジェネリック・データ・モード

このモードではサブネットワーク・ノードと ABC 間のマスター-スレーブ関係はありません。ABC も含むサブネットワーク上のどのノードも自然にメッセージをプロデュース / コンsums します。

2-5 “ジェネリック・データ・モード” を参照してください。

### プロトコル構築ブロック

次の構築ブロックはサブネットワーク通信を記述するために ABC コンフィグツールで使用されます。これらのブロックがいかに 2 つのプロトコル・モードに適応するかこのドキュメントで後述します。

- ノード

サブネットワーク上の 1 つの機器のノードを表します。各ノードはトランザクションの番号に関連付けられます。下を参照ください。

- トランザクション

トランザクションは完全なシリアル・テレグラムを表し、フレームオブジェクト（下記参照してください）の番号で構成されます。各トランザクションはサブネットワーク上でいかに、そしていつ使用するかを制御するパラメータのセットに関連付けられています。

- コマンド

‘コマンド’は ABC コンフィグツールに格納された事前定義されたトランザクションです。格納され再利用されたトランザクションを使用することで共通操作を単純化します。

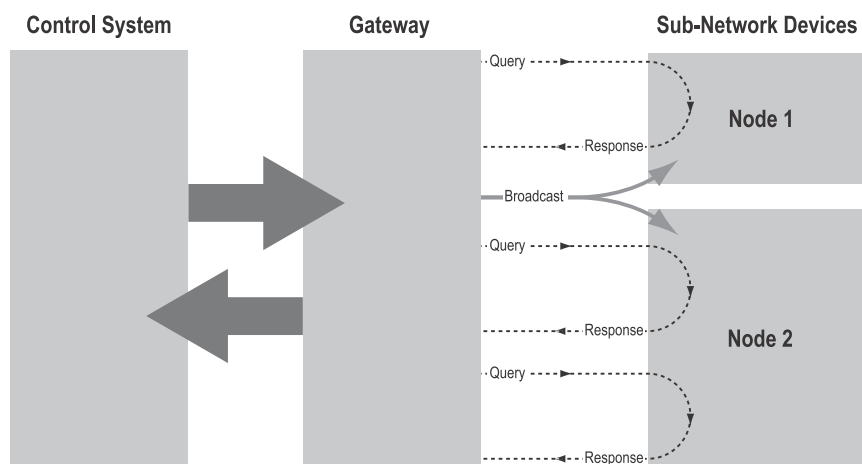
- フレーム・オブジェクト

‘フレーム・オブジェクト’はトランザクションを作成するために使用されるローレベル・エンティティです（上記参照してください）。フレーム・オブジェクトは固定された値（常に固定）、値の範囲（制限オブジェクト）、データ・ブロックあるいは計算されたチェックサムを表します。

## マスター・モード

このモードでは通信はクエリ / レスポンス・スキームをベースにしています。ABC がサブネットワーク上でクエリを発行するとクエリを受けたノードはレスポンスを返します。ノードはクエリの受信をすることなく自発的にレスポンスすることができません。

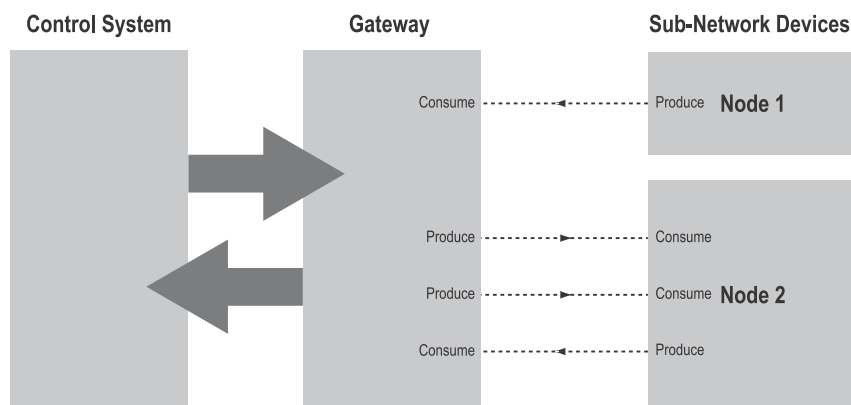
このルールには一つ例外があり、それがブロードキャストです。ほとんどのプロトコルはネットワーク上の全てのノードに対して応答を必要としないブロードキャスト・メッセージを送信する方法を提供します。これはブロードキャスト・ノードと通信する機能がある ABC でも使用されます。



マスター・モードでは ABC コンフィグツールはほとんど共通して Modbus RTU コマンドを使用して事前にロードされ、ABC コンフィグツールでノードを右クリックすることで立ち上がり、'Insert New Command' を選択します。しかしこれは同じクエリ / レスポンス・メッセージ・スキームに基づく他のプロトコルの実装の妨げになりません。

## ジェネリック・データ・モード

このモードではサブネットワーク上のノードと ABC 間ではマスター / スレーブの関係はありません。ABC 自身も含むどのノードも自然にメッセージをプロデュース / コンsumします。ノードはメッセージに応答する必要はなくまた送信するためにクエリを待つ必要もありません。



上の図で ABC はサブネットワーク上のノードにより、'プロデュース'されたデータを、'コンsum'しています。この'コンsum'データは上位レベル・ネットワークからアクセスできます。これはまた別の経路では向きを変えます：上位レベル・ネットワークから受信されたデータは、サブネットワーク上でノードによって、'コンsum'されるメッセージの、'プロデュース'に使用されます。

# EtherNet/IP

## 概論

EtherNet/IP は ControlNet と Ethernet/IP のアプリケーション層でもある制御と情報プロトコル (CIP) に基づいています。ABC は EtherNet/IP ネットワーク上でグループ 2 と 3 サーバとして動作します。

入力と出力データは I/O 接続または Assembly オブジェクトと Parameter Input/Output Mapping オブジェクトへのエクスプリシット・メッセージを使用しアクセスされます。

- 21-1 “CIP オブジェクトの実装” も参照してください

## データ・タイプ

入力と出力は I/O データとパラメータ・データ、二つのデータ・タイプをもちます。値の変更があったときに I/O データが交換され、Assembly オブジェクトへの I/O 接続を使用しアクセスできます。

パラメータ・データはアサイクリックに Parameter Input/Output Mapping オブジェクトを介しアクセスすることができます。これらのオブジェクト内の各インスタンス属性は ABC コンフィグツールを使用し手動で作成されなければなりません。詳細については、A-1 “パラメータ・データの初期化 (エクスプリシット・データ)” を参照してください。

- 21-4 “アセンブリ・オブジェクト、クラス 04h”
- 21-6 “パラメータ・データ入力マッピング・オブジェクト、クラス B0h”
- 21-7 “パラメータ・データ出力マッピング・オブジェクト、クラス B1h”
- 11-1 “フィールドバス設定”

も参照してください。

## メモリ・レイアウト

入力 / 出力データ領域のデータは次のように継続的な I/O データ・ブロックとして表示されます。

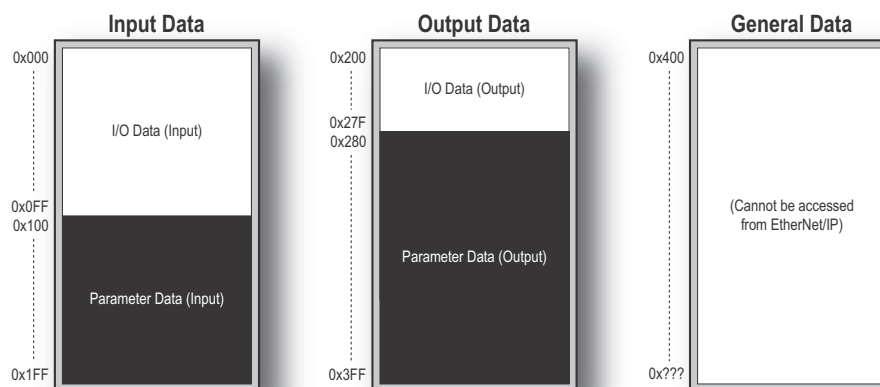
**例:**

この例での ABC の I/O データの値は次の通りに設定されています。

IO サイズ入力 = 256 バイト (0x0100)

IO サイズ出力 = 64 バイト (0x0040)

メモリ・レイアウトの結果:





# Modbus/TCP

## 概論

Modbus/TCP プロトコルは TCP/IP の最上位で動作する標準 Modbus プロトコルの実装です。Modbus/TCP サーバ内の実装は Modbus/TCP 仕様内定義された機能のサブネットを介し入力 / 出力データ領域へのアクセスを提供します。

サーバは TCP ポートを介し、8 個までの接続と通信をサポートします。Modbus/TCP プロトコルの詳細については Open Modbus 仕様を参照してください。

## アドレッシング・モード

ABC は Modbus 通信において二つの異なる操作モードを使用します。

- **Modbus アドレッシング・モード (デフォルト)**

このモードでは入力 / 出力データ領域は異なる機能コードにマッピングされます。コイルのアドレッシングはこのモードでは不可能なことに注意してください。

- 2-8 “Modbus アドレッシング・モード” も参照してください。

- **Anybus アドレッシング・モード**

Modbus アドレッシング・モードと比較して柔軟な方法でデータをアドレスします。一方複数の機能コードは同じ ABC のデータにアクセスするために使用できます。最初は混乱するかもしれませんが Modbus アドレッシング・コードでは不可能な操作を可能にします。(同じメモリロケーションに関連したコイルへのアクセスによるレジスタによるビット毎の操作を可能にします。)

- 2-9 “Anybus アドレッシング・モード” も参照してください。

## サポートされている例外コード

コード	名前	説明
0x01	不正な機能	クエリ内の機能コードはサポートされていません。
0x02	不正なデータ・アドレス	クエリ内で受信されたデータ・アドレスは初期化されたメモリ領域外です。
0x03	不正なデータ値	要求されたデータは不正です。

## Modbus アドレッシング・モード

### サポートされている機能コード

次の機能コードがこのモードで使用することができます。

Modbus 機能	機能コード	関連する領域
Read Holding Registers	3	出力データ領域 (0x200...0x3FF)
Read Input Registers	4	入力データ領域 (0x000...0x1FF)
Write Single Register	6	出力データ領域 (0x200...0x3FF)
Force Multiple Registers	16	出力データ領域 (0x200...0x3FF)
Mask Write Register	22	出力データ領域 (0x200...0x3FF)
Read/Write Registers	23	出力データ領域 (0x200...0x3FF)

### 入力レジスタ・マップ

入力データ領域は次のように入力レジスタにマッピングされます：

レジスタ #	ABC のメモリロケーション	説明
1	0x000... 0x001	各レジスタは入力データ領域内で 2 バイトに対応
2	0x002... 0x003	
3	0x004... 0x005	
4	0x006... 0x007	
5	0x008... 0x009	
6	0x00A... 0x00B	
...	...	
255	0x1FC... 0x1FD	
256	0x1FE... 0x1FF	

### 保持レジスタ・マップ

出力データ領域は次のように保持レジスタにマッピングされます。

レジスタ #	ABC のメモリロケーション	説明
1	0x200... 0x201	各レジスタは出力データ領域内で 2 バイトに対応
2	0x202... 0x203	
3	0x204... 0x205	
4	0x206... 0x207	
5	0x208... 0x209	
6	0x20A... 0x20B	
...	...	
255	0x3FC... 0x3FD	
256	0x3FE... 0x3FF	

## Anybus アドレッシング・モード

### サポートされている機能コード

次の機能コードがこのモードで使用することができます。

Modbus 機能	機能コード	関連する領域
Read Coil	1	入力 / 出力データ領域 (0x000... 0x3FF)
Read Input Discretes	2	
Read Holding Registers	3	
Read Input Registers	4	
Write Coil	5	出力データ領域 (0x200... 0x3FF)
Write Single Register	6	
Force Multiple Coils	15	
Force Multiple Registers	16	
Mask Write Register	22	
Read/Write Registers	23	入力 / 出力データ領域 (0x000... 0x3FF)

### コイル & レジスタ・マップ

入力 / 出力データ領域はコイルとレジスタに次のようにマッピングされます。

レジスタ #	コイル #	ABC のメモリロケーション	領域	説明
1	1... 16	0x000... 0x001	入力データ領域	-
2	17... 32	0x002... 0x003		
3	33... 48	0x004... 0x005		
4	49... 64	0x006... 0x007		
...	...	...		
255	4065... 4080	0x1FC... 0x1FD		
256	4081... 4096	0x1FE... 0x1FF		
257	4097... 4112	-	-	(リザーブド)
...	...	-		
1024	16369... 16384	-	出力データ領域	-
1025	16385... 16400	0x200... 0x201		
1026	16401... 16416	0x202... 0x203		
1027	16417... 16432	0x204... 0x205		
1028	16433... 16448	0x206... 0x207		
...	...	...		
1279	20449... 20464	0x3FC... 0x3FD		
1280	20465... 20480	0x3FE... 0x3FF		

**注意 1：**上記の表は全ての機能コードに適用されます。

**注意 2：**コイルは最初 MSB にマッピングされます。コイル 0 はレジスタ 0 のビット 15 に対応します。

# ファイル・システム

## 概論

### 概論

ABC には Web ファイル、ネットワーク通信、E-mail メッセージ等の情報を搭載するために使用されるファイル・システムが組み込まれています。

### 慣例

- ‘¥’ (バックスラッシュ) はパスのセパレータとして使用されます。
- ‘パス’ はシステム・ルートから始まり、‘¥’ で開始されなければなりません。
- ‘パス’ は ‘¥’ で終了してはなりません。
- 名前はスペース (‘ ’) を含むことができますが、始まりと終わりでは使用することができません。
- 名前には次の文字を使用することができません。‘¥ / : \* ? “ < > | ’
- 名前は 48 文字を超えることはできません。(プラス nul 終端)
- パスは 256 文字を超えることはできません。(ファイル名を含む)
- 同時に開くことができるファイル数は 40 です。
- 同時に開くことができるディレクトリは 40 です。

### 重要な注意：

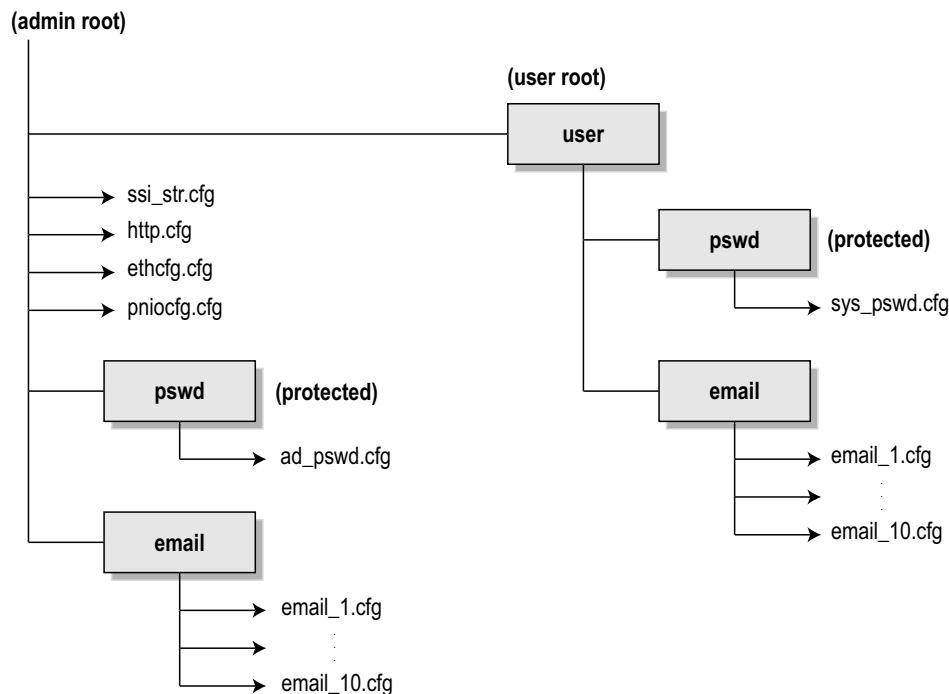
不揮発性ストレージは FLASH メモリにあります。各 FLASH セグメントはこのメモリの種類の性質からおよそ 1,000,000 回消去することができます。

次の操作で一つ以上の FLASH セグメントを消去します。

- ファイルまたはディレクトリの消去、移動または名前の変更
- 書き込みまたは既存のファイルへのデータの追加
- ファイル・システムのフォーマット
- スキャナ・コンフィグレーションの保存

## 概要

下の図は一般 / アドミニ・ユーザがアクセスすることのできる領域のファイル・システムの構成を示しています。



## システム・ファイル

ファイル・システムはシステム・コンフィグレーションのために使用される一連のファイルを含みます。”system files”として知られるこれらのファイルは正規の ASCII ファイルで、標準テキスト・エディタ（Microsoft Windows™ のノートパッドのような）を使用して変更することができます。これらのファイルのいくつかは SSI を使用する時 ABC でも変更することができます。（8-1 “Server Side Include (SSI)” を参照してください。）

システム・ファイルのフォーマットは値が割り付けられている各 ‘keys’ のコンセプトに基づいています。次の例を参照してください。

**例:**

```
[Key1]
value of key1

[Key2]
value of key2
```

各システム・ファイルの正確なフォーマットの詳細は本ドキュメントに後述されます。

# 基本的なネットワーク・コンフィグレーション

## 概論

ABC はネットワーク設定について三つの操作モードを提供します：

- **コンフィグレーション・スイッチにより指定された設定**

オンボード・スイッチが 0 以外の値に設定されている場合、ABC 次の設定に固定されます。

IP アドレス：	192.168.0.x	(x = スイッチ値)
ゲートウェイ：	255.255.255.0	
サブネット：	255.255.255.0	
DHCP:	オフ	

詳細については 1-4 “コンフィグレーション・スイッチ” を参照してください。

- **ABC コンフィグツールで指定された設定**

オンボード・スイッチが 0 に設定されていて ABC コンフィグツールで設定がされている場合 (“TCP/IP Settings”=enabled)、ABC は ABC コンフィグツールで作成されたコンフィグレーションからの設定を使用します。

このような場合、次の動作が原因でシステム・ファイルの内容 ‘ethcfg.cfg’ は完全に無視されます：

- DNS サービスが利用できない
- ドメインとホスト名が設定できない
- Email サービスが利用できない
- ネットワークから受信した設定（つまり HICP または DCP）が電源不足またはリセットのイベントによって失われる

- **‘ethcfg.cfg’ 内で指定された設定**

オンボード・スイッチが 0 に設定されていて ABC コンフィグツールで設定がされていない場合 (“TCP/IP Settings”=disabled)、ABC はシステム・ファイル ‘ethcfg.cfg’ に搭載された設定を使用します。

このファイルがない場合、30 秒間 ABC は DHCP または HICP を介し設定を読み込むを試みます。この間にコンフィグレーションがない場合、ABC は停止しオンボード LED にエラー表示します。

### EtherNet/IP

TCP/IP 設定は TCP/IP インターフェース・オブジェクトを介して EtherNet/IP からアクセスされます。

- 21-9 “TCP/IP インターフェース・オブジェクト、クラス F5h” も参照してください。

### DHCP/BootP

ABC は DHCP または BootP サーバから TCP/IP 設定を読み込みます。DHCP サーバが見つからない場合は、ABC は現在の設定を使用します（現在 ‘ethcfg.cfg’ に保存されている設定）。

現在の設定が使用できない場合（つまり、‘ethcfg.cfg’ がいない、または不正な設定が含まれている）、ABC は停止しオンボードステータス LED にエラー表示します。（ネットワーク・コンフィグレーションはしかしながら HICP でアクセスできます。

4-4 “Anybus IPconfig (HICP)” を参照してください。

## Ethernet コンフィグレーション・ファイル ('ethcfg.cfg')

### 概論

ネットワークに接続するためにABCは有効なTCP/IP コンフィグレーションを必要とします。これらの設定はシステム・ファイル 'ethcfg.cfg' に保存されます。

#### ファイル・フォーマット

[IP address] xxx.xxx.xxx.xxx	• IP アドレス
[Subnet mask] xxx.xxx.xxx.xxx	
[Gateway address] xxx.xxx.xxx.xxx	• サブネット・マスク
[DHCP/BOOTP] ON or OFF	• ゲートウェイ・アドレス
[SMTP address] xxx.xxx.xxx.xxx	• DHCP/BootP
[SMTP username] username	
[SMTP password] password	
	ON - 有効 OFF - 無効
	• SMTP サーバ / ログイン設定
	ユーザ名とパスワードはサーバによって要求されたときのみ必要です。
[DNS1 address] xxx.xxx.xxx.xxx	• プライマリとセカンダリ DNS
[DNS2 address] xxx.xxx.xxx.xxx	
	ホスト名の決定に必要です。
[Domain name] domain	• 条件付きでないホスト名のためのデフォルト・ドメイン名
[Host name] anybus	• ホスト名
[HICP password] password	• HICP パスワード

このファイルの設定は次により影響します。

- EtherNet/IP (4-1 “EtherNet/IP” を参照してください)
- HICP (4-4 “Anybus IPconfig (HICP)” を参照してください)
- SSI (8-1 “Server Side Include (SSI)” を参照してください)
- 5-1 “FTP サーバ”
- 11-1 “フィールドバス設定”

も参照してください

## IP アクセス制御

ABC への接続を許可された IP アドレスを指定することができます。この情報はシステム・ファイル '¥ip\_accs.cfg' に保存されます。

### ファイル・フォーマット：

[Web] xxx.xxx.xxx.xxx	• ウェブ・サーバにアクセスできるノードがここにリストされます
[FTP] xxx.xxx.xxx.xxx	• FTP サーバにアクセスできるノードがここにリストされます
[Modbus/TCP] xxx.xxx.xxx.xxx	• Modbus/TCP を介し ABC にアクセスできるノードがここにリストされます
[EtherNet/IP] xxx.xxx.xxx.xxx	• EtherNet/IP を介し ABC にアクセスできるノードがここにリストされます
[All] xxx.xxx.xxx.xxx	• ひとつまたは複数の上記の key が省略される場合 ABC によって使用されるフォールバック設定

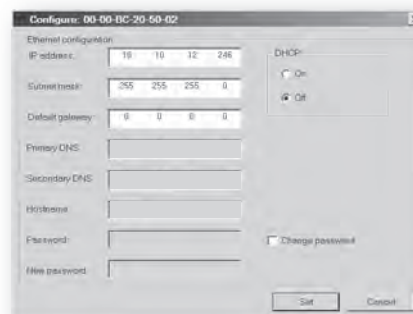
**注意：**\* は IP シリーズを選択するときにワイルドカードとして使用できます



## Anybus IPconfig (HICP)

ABC は HMS のウェブサイトから無償でダウンロードできる Anybus IPconfig ユーティリティによって使用される HICP プロトコルをサポートしています。このユーティリティはどのような Anybus 製品が接続されたネットワーク設定にも使用できます。正常であれば、コンフィグレーション・ファイルの設定と置き換わります ('ethcfg.cfg')。

プログラムの開始で、ネットワークは Anybus 製品のためにスキャンされます。'Scan' をクリックすることでネットワークはいつでも再スキャンされます。検出されたデバイスのリストで、ABC アダプタは 'ABC-PRT' として表示されます。ネットワーク設定を変更するためにはリスト内のエントリをダブルクリックします。



IP コンフィグレーションとパスワード設定が含まれたウィンドウが現れます。新しい設定を有効にするために 'Set' をクリックし、中止するために 'Cancel' をクリックします。

コンフィグレーションはパスワードによってプロテクトすることもできます。パスワードを入力するために 'Change password' をクリックし、'New password' の下にパスワードを入力します。プロテクトされるとコンフィグレーションを変更するためにユーザは有効なパスワードを要求されます。

'Set' をクリックすると新しい IP コンフィグレーションはコンフィグレーション・ファイル ('ethcfg.cfg') に保存されます。

ABC コンフィグツールの TCP/IP 設定が有効になっている場合、HICP を介して受信した設定は電源ロスまたはリセットのイベントにより失われます。

# FTP サーバ

## 概論

FTP サーバ・ビルトインは標準 FTP クライアントを使用してファイル・システムにアクセスします。

次のポート番号は FTP 通信のために使用されます。

- TCP, port 20 (FTP data port)
- TCP, port 21 (FTP command port)

### セキュリティ・レベル

FTP サーバはアドミニと一般、二つのセキュリティ・レベルを使用します。

- 一般レベル・ユーザ  
ルート・ディレクトリは 'user'
- アドミニ・レベル・ユーザ  
ルート・ディレクトリは 'x' でユーザはファイル・システムへのアクセスに制限がありません。

### ユーザ・アカウント

ユーザ・アカウントは二つのファイルに保存され、Web からのアクセスをプロテクトします：

- 'userpswdsys\_pswd.cfg'  
ファイルは一般ユーザ向けのユーザ・アカウントをもちます。
- 'pswdad\_pswd.cfg'  
ファイルはアドミニ向けのユーザ・アカウントをもちます。

#### ファイル・フォーマット：

これらのファイルのフォーマットは次の通りです：

```
Username1:Password1
Username2:Password2
Username3:Password3
```

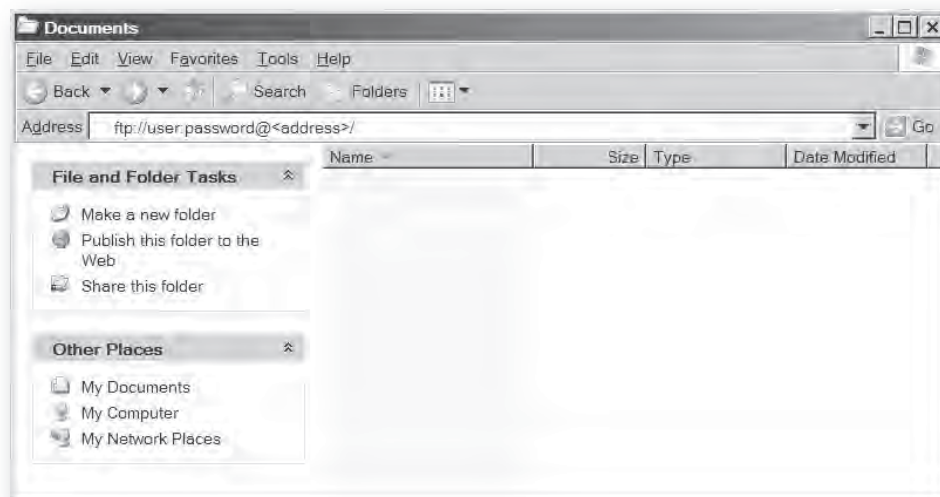
**注意 1：**有効なユーザ・アカウントが定義されていない場合、ABC はアドミニによるアクセスを全てのユーザに与えます。この場合、FTP はどのようなユーザ名 / パスワードの組み合わせも受け入れ、ルート・ディレクトリは 'x' になります。

**注意 2：**FTP サーバはユーザ・アカウントに Telnet サーバを与えます。

## FTP 接続例（Windows エクスプローラ）

Windows エクスプローラの FTP クライアント・ビルトインはファイル・システムにアクセスするために次のように簡単に使用できます：

1. ‘Start’ ボタン上で右クリックし Windows エクスプローラを開き、‘Explore’ を選択します。
2. アドレスフィールドで、FTP://<user>:<password>@<address> を入力します。
  - <address> に ABC の IP アドレスを入力します
  - <user> にユーザ名を入力します
  - <password> にパスワードを入力します。
3. Enter を押します。エクスプローラは指定した設定を使用している ABC への接続を促します。成功するとエクスプローラ・ウィンドウのファイル・システムが表示されます。



# Telnet サーバ

## 概論

Telnet サーバのビルトインは標準 Telnet クライアントを使用してファイル・システムへアクセスします。サーバは TCP ポート 23 を介して通信します。

### セキュリティ・レベル

FTP サーバのように Telnet サーバはアドミニと一般、二つのセキュリティ・レベルを使用します。

- 一般レベル・ユーザ

ルート・ディレクトリは 'user'

- アドミニ・レベル・ユーザ

ルート・ディレクトリは 'x' でユーザはファイル・システムへのアクセスに制限がありません。

### ユーザ・アカウント

Telnet サーバは FTP サーバのアカウントを与えます。有効なユーザ・アカウントが定義されない場合、ABC はアドミニによるアクセスを全てのユーザに与えます。この場合、ログインする必要はなく、ルート・ディレクトリは 'x' になります。

詳細については 5-1 “ユーザ・アカウント” を参照してください。

## 一般的なコマンド

### admin

- 構文  
admin
- 説明  
ユーザは有効なアドミンのユーザ名 / パスワードの組み合わせを提供することができ、このコマンドは一般レベル・ユーザへのアドミニ・アクセス権を提供します。

### exit

- 構文  
exit
- 説明  
このコマンドは Telnet セッションをクローズします。

### help

- 構文  
help [general|diagnostic|filesystem]
- 説明  
アーギュメントが指定されない場合、次のメニューが表示されます。

Genaral commands:

help	- Help with menus
version	- Display version information
exit	- Exit station program

Also try 'help [general|diagnostic|filesystem]'

### version

- 構文  
version
- 説明  
このコマンドはコミュニケータ内の Ethernet モジュールのバージョン情報、シリアル番号と MAC ID を表示します。

## 診断コマンド

### arp

- 構文  
arp
- 説明  
ARP ステートとテーブルを表示します。

### iface

- 構文  
iface
- 説明  
ネット・インターフェース・ステートを表示します。

### routes

- 構文  
routes
- 説明  
IP ルート表を表示します。

### sockets

- 構文  
sockets
- 説明  
ソケットリストを表示します。

## ファイル・システム操作

ファイル名、ディレクトリ名またはパスのコマンドはアーギュメントとして与えられ名前は直接または括弧内に書き込まれます。スペースを含む名前はファイル名が括弧で囲まれていなければなりません。‘?’、‘\’ と ‘.’ に関連したパス名を使用することも可能です。

### append

- 構文  
append [file] ["The line to append"]
- 説明  
ファイルに行を追加します。

**cd**

- **構文**  
`cd [path]`
- **説明**  
現在のディレクトリを変更します。

**copy**

- **構文**  
`copy [source] [destination]`
- **説明**  
このコマンドは指定した場所のソース・ファイルのコピーを作成します。

**del**

- **構文**  
`del [file]`
- **説明**  
ファイルを削除します。

**dir**

- **構文**  
`dir [path]`
- **説明**  
ディレクトリの内容をリストします。パスが与えられていない場合、現在のディレクトリの内容がリストされます。

**df**

- **構文**  
`df`
- **説明**  
ファイル・システム情報を表示します。

**format**

- **構文**  
`format`
- **説明**  
ファイル・システムをフォーマットします。これは特権的なコマンドでアドミニストレーション・モードのとき呼び出されます。

**md**

- **構文**  
`md [directory]`
- **説明**  
ディレクトリを作成します。パスが与えられていない場合、ディレクトリは現在のディレクトリ内に作成されます。

**mkfile**

- **構文**  
`mkfile [filename]`
- **説明**  
空のファイルを作成します。

**move**

- **構文**  
`move [source] [destination]`
- **説明**  
このコマンドはソースの場所からファイルまたはディレクトリを指定した方向に移動します。

**rd**

- **構文**  
`rd [directory]`
- **説明**  
ディレクトリを削除します。空のときのみディレクトリは削除できます。

**ren**

- **構文**  
`ren [old name] [new name]`
- **説明**  
ファイルまたはディレクトリをリネームします。

**type**

- **構文**  
`type [filename]`
- **説明**  
ファイルの内容を入力します。



# Web サーバ

## 概論

ABC は SSI 機能をもった柔軟な Web サーバを使用します。ビルトイン Web ページは特殊なアプリケーションに合わせるためや I/O データとコンフィグレーション設定にアクセスするためにカスタマイズが可能です。

Web サーバはポート 80 を介して通信します。

- 8-1 “Server Side Include (SSI)”
- 4-3 “IP アクセス制御”

も参照してください。

### プロテクトされたファイル

セキュリティ上の理由から、以下のファイルはウェブアクセスからプロテクトされます：

- ‘¥user¥pswd’ にあるファイル’
- ‘¥pswd’ にあるファイル
- ファイル名 ‘web\_accs.cfg’ を含むディレクトリ内にあるファイル

### デフォルト Web ページ

ABC はネットワーク・パラメータのコンフィグレーションのための Web ページを作成するための仮想ファイル一式を含みます。これらの仮想ファイルはディスク 0 のルート内に同じファイル名で上書き（消去不可）できます。

これにより例えば HMS ロゴを ‘¥logo.jpg’ という名前の新しいロゴに置き換えることができます。仮想コンフィグレーション・ページへ Web ページからのリンクを作成することもできます。この場合、リンクは ‘¥config.htm’ に向いている必要があります。

これらの仮想ファイルは：

¥index.htm	- Points to the contents of config.htm
¥config.htm	- Configuration frame page
¥configform.htm	- Configuration form page
¥configform2.htm	- Configuration form page
¥store.htm	- Configuration store page
¥logo.jpg	- HMS logo
¥configuration.gif	- Configuration picture
¥boarder.bg.gif	- picture
¥boarder_m_bg.gif	- picture

## 許可

ディレクトリは 'web\_accs.cfg' という名前のファイルをディレクトリに置くことによって Web アクセスから守ることができます。このファイルはディレクトリとサブディレクトリへのアクセスが許可されたユーザのリストを含んでいます。

### ファイル・フォーマット

```
Username1:Password1
Username2:Password2
...
UsernameN:PasswordN
```

承認されたユーザのリスト

```
[AuthName]
(message goes here)
```

オプションでログインメッセージは key [AuthName](メッセージはここに来ます) を含むことで指定されます。このメッセージはプロテクトされたディレクトリにアクセスする Web ブラウザで表示されます

承認されたユーザのリストはひとつまたは複数のファイルにリダイレクションできます。

### 例:

この例では、'here.cfg' と 'too.cfg' のファイルから承認されたリストがロードされます。

```
[File path]
¥i¥put¥it¥over¥here.cfg
¥i¥actually¥put¥some¥of¥it¥over¥here¥too.cfg
```

```
[AuthName]
Yeah. Whatsda passwoid?
```

この機能を使用する場合、ユーザ / パスワード・ファイルを Web アクセスからプロテクトされたディレクトリに置くことに注意してください。7-1 “プロテクトされたファイル” を参照してください。

# 内容の種類

デフォルトでは、次の内容の種類がファイルの拡張子によって認識されます。

内容の種類	ファイルの拡張子
text/html	*.htm, *.html, *.shtm
image/gif	*.gif
image/jpeg	*.jpeg, *.jpg, *.jpe
image/x-png	*.png
application/x-javascript	*.js
text/plain	*.bat, *.txt, *.c, *.h, *.cpp, *.hpp
application/x-zip-compressed	*.zip
application/octet-stream	*.exe, *.com
text/vnd.wap.wml	*.wml
application/vnd.wap.wmlc	*.wmlc
image/vnd.wap.wbmp	*.wbmp
text/vnd.wap.wmlscript	*.wmls
application/vnd.wap.wmlscriptc	*.wmlsc
text/xml	*.xml
application/pdf	*.pdf

レポートされた内容の種類と SSI でスキャンされるファイルをコンフィグレーション / 再コンフィグレーションできます。これはシステム・ファイル ‘\$http.cfg’ の中で行われます。

## ファイル・フォーマット

```
[FileTypes]
FileType1:ContentType1
FileType2:ContentType2
...
FileTypeN:ContentTypeN

[SSIFileTypes]
FileType1
FileType2
...
FileTypeN
```

**注意：** 50 の内容の種類と 50 の SSI ファイルの種類までこのファイルで指定できます。

# Server Side Include (SSI)

## 概論

Server Side Include（以下、SSI と表記します）機能は、動的な内容が Web ページと Email メッセージ内での使用を可能にします。

SSI はソース・ドキュメント内に組み込まれた特別なコマンドです。ABC はこのようなコマンドを検出するとそれを実行し、（可能な場合）結果と置き換えます。

### シンタックス

下の 'X' はコマンドに関連したコマンド OP コードとパラメータを表します。

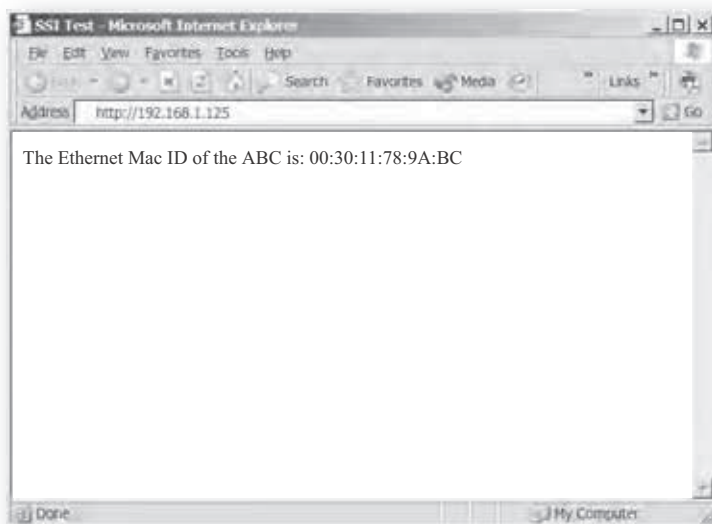
```
<?--#exec cmd_argument='XXXXXXXXXXXXXXXXXXXXXXX'-->
```

### 例

次の例は ABC の Ethernet MAC ID を表示するための Web ページに関連します。

```
<HTML>
<HEAD><TITLE>SSI Test</TITLE></HEAD>
<BODY>
The Ethernet Mac ID of the ABC is:
<?--#exec cmd_argument='DisplayMacID'-->
</BODY>
</HTML>
```

結果のウェブページ：



## 機能

### DisplayMacID

この機能はフォーマット xx:xx:xx:xx:xx:xx で MAC ID を返します。

構文:

```
<?--#exec cmd_argument='DisplayMacId'-->
```

### DisplaySerial

この機能はネットワーク・インターフェースのシリアル番号を返します。

構文:

```
<?--#exec cmd_argument='DisplaySerial'-->
```

### DisplayFWVersion

この機能はネットワーク・インターフェースのメインのファームウェア・リビジョンを返します。

構文:

```
<?--#exec cmd_argument='DisplayFWVersion'-->
```

### DisplayBLVersion

この機能はネットワーク・インターフェースのブートローダ・リビジョンを返します。

構文:

```
<?--#exec cmd_argument='DisplayBLVersion'-->
```

### DisplayIP

この機能は現在使用されている IP アドレスを返します。

構文:

```
<?--#exec cmd_argument='DisplayIP'-->
```

### DisplaySubnet

この機能は現在使用されているサブネット・マスクを返します。

構文:

```
<?--#exec cmd_argument='DisplaySubnet'-->
```

### DisplayGateway

この機能は現在使用されているゲートウェイ・アドレスを返します。

構文:

```
<?--#exec cmd_argument='DisplayGateway'-->
```

### DisplayDNS1

この機能はプライマリ DNS サーバのアドレスを返します。

**構文:**

```
<?--#exec cmd_argument='DisplayDNS1'-->
```

### DisplayDNS2

この機能はセカンダリ DNS サーバのアドレスを返します。

**構文:**

```
<?--#exec cmd_argument='DisplayDNS2'-->
```

### DisplayHostName

この機能はホスト名を返します。

**構文:**

```
<?--#exec cmd_argument='DisplayHostName'-->
```

### DisplayDomainName

この機能はデフォルト・ドメイン名を返します。

**構文:**

```
<?--#exec cmd_argument='DisplayDomainName'-->
```

### DisplayDhcpState()

この機能は DHCP/BootP が有効か無効かを返します。

**構文:**

```
<?--#exec cmd_argument='DisplayDhcpState( "Output when ON", "Output when OFF" )'-->
```

### DisplayDhcpSupport()

この機能は DHCP がサポートされている場合、‘Arg1’ を、サポートされていない場合 ‘Arg2’ を返します。

**構文:**

```
<?--#exec cmd_argument='DisplayDhcpSupport( "Arg1", "Arg2" )'-->
```

### DisplayEmailServer

この機能は現在使用されている SMTP サーバアドレスを返します。

**構文:**

```
<?--#exec cmd_argument='DisplayEmailServer'-->
```

### DisplaySMTPUser

この機能は SMTP 証明のために使用されるユーザ名を返します。

**構文:**

```
<?--#exec cmd_argument='DisplaySMTPUser'-->
```

### DisplaySMTPPwd

この機能は SMTP 証明のために使用されるパスワードを返します。

**構文:**

```
<?--#exec cmd_argument='DisplaySMTPPwd'-->
```

### StoreEtnConfig

**注意:** この機能は Email メッセージ内では使用できません。

この機能はコンフィグレーション・ファイル 'ethcfg.cfg' 内のパスした IP コンフィグレーションを保存します。

**構文:**

```
<?--#exec cmd_argument='StoreEtnConfig'-->
```

HTML ページのこの行が含まれ新しい IP 設定を使用したフォームを渡します。

**フォーム内の受け入れられるフィールド:**

```
SetIp
SetSubnet
SetGateway
SetEmailServer
SetDhcpState - value "on" or "off"
SetDNS1
SetDNS2
SetHostName
SetDomainName
SetSMTPUser
SetSMTPPwd
```

**デフォルトの出力:**

```
Invalid IP address!
Invalid Subnet mask!
Invalid Gateway address!
Invalid IP address or Subnet mask!
Invalid Email Server IP address!
Invalid DHCP state!
Invalid DNS1!
Invalid DNS2!
Configuration stored correctly.
Failed to store configuration.
```

## GetText()

**注意：**この機能は Email メッセージ内では使用できません。

この機能はオブジェクトからテキスト・ストリング・フォームを集め出力データ領域に保存します。

### 構文：

```
<?--#exec cmd_argument='GetText( "ObjName", OutWriteString ( offset ), n )'-->
```

ObjName	– オブジェクト名
offset	– 出力データ領域の最初からオフセットを指定します。
n	– 読み込む文字の最大数を指定します（選択機能）

### デフォルト出力：

Success	– Write succeeded
Failure	– Write failed

## printf()

この機能は Web ページ上の入力 / 出力データ領域からのデータを含むフォーマットされたストリングを含みます。ストリングのフォーマットはC言語機能のprintf() と似ています。

### 構文：

```
<?--#exec cmd_argument='printf("String to write", Arg1, Arg2, ..., ArgN)'-->
```

C 言語機能 printf() のように、SSI 機能のための "String to write" は二つのオブジェクトの種類が含まれます：出力ストリームへコピーされる通常の文字と変換仕様は printf への連続するアーギュメントの変換とプリントをさせます。各変換仕様は文字%で開始し変換文字で終了します。%と変換文字の間は順番に：

- 仕様を変更するフラグ（一定の順番）
  - フィールド内の変換されたアーギュメントの左の調整することを指定します。
  - + 番号は常に記号でプリントされることを指定します。
  - (space) 最初の文字が記号ではない場合、スペースが接頭辞になります。
  - 0 数字の変換のために、ゼロで始まるフィールドに挿入句を指定します。
  - # 交互の出力フォームを指定します。o のために、最初の数字はゼロになります。x または X、0x または 0X はゼロ以外の結果の接頭辞になります。e、E、f、g、G、出力は常に十進点をもちます；g と G、続くゼロは削除できません。
- 最小フィールド幅を指定する番号です。変換されたアーギュメントはフィールド内で最低この幅が必要な場合それ以上でプリントされます。変換されたアーギュメントはフィールド幅よりも少ない文字をもち、フィールド幅を確保するため左に詰められます（または右、左の調整が要求された場合）。補充文字は通常スペースですが、0 がフラグにある場合、代用することもできます。
- ピリオドは精密にフィールド幅を分けます。
- 番号、精度は e、E または F 変換からプリントされるストリング、または数字の数字から、g、または G 変換にのための重要な数字の番号、または整数のためにプリントされる数字の最小番号によってプリントされる文字の最大数を指定します（最初の 0 は必要な幅を作るために追加されます。）
- 長さの修飾語 h、l（文字 ell）、または L。"h" は関連するアーギュメントが short または unsigned short でプリントされることを示します、"l" はアーギュメントが long または unsigned longであることを示します。



変換文字と意味は以下に示す通りです。% の後の文字が変換文字でない場合、動作は設定されません。

文字	アーギュメント・タイプ、変換結果
d, i	バイト、ショート、十進数表記 (signed 表記は signed アーギュメントを使用)
o	バイト、ショート、八進数表記 (先頭のゼロなし)
x, X	バイト、ショート、16 進数表記 (先頭 0x または 0X)、0x のために abcdef または 0X のために ABCDEF を使用
u	バイト、ショート; 十進数表記
c	バイト、ショート、シングルの文字; unsigned char へ変換後
s	char*; スtringからの文字は "%0" までまたは精度で示された文字数がプリントされます
f	float; 精度によって d's の数が指定されたフォーム [-]mmm.ddd の十進数表記です。デフォルトの精度は 6; 0 の精度は十進数点を省略します
e, E	float; 精度によって d's の数が指定されたフォーム [-]m.dddddd e+-xx または [-]m.ddddddE+-xx の十進数表記です。デフォルトの精度は 6; 0 の精度は十進数点を省略します。
g, G	G float; べき指数が -4 より小さいか、精度以上の場合 %e または %E が使用されます; さもないと %f が使用されます。続くゼロと十進数点はプリントされません。
%	アーギュメントは変換されません; a% がプリントされます

SSI 機能 *printf* に送られるアーギュメントは:

アーギュメント	説明
InReadSByte( <i>offset</i> )	入力データ領域のオフセット位置から Signed バイトの読み込み
InReadUByte( <i>offset</i> )	入力データ領域のオフセット位置から Unsigned バイトの読み込み
InReadSWord( <i>offset</i> )	入力データ領域のオフセット位置から Signed ワードの読み込み
InReadUWord( <i>offset</i> )	入力データ領域のオフセット位置から Unsigned ワードの読み込み
InReadSLong( <i>offset</i> )	入力データ領域のオフセット位置から Signed ワードの読み込み
InReadULong( <i>offset</i> )	入力データ領域のオフセット位置から long ワードの読み込み
InReadString( <i>offset</i> )	入力データ領域のオフセット位置から String (char*) の読み込み
InReadFloat( <i>offset</i> )	入力データ領域のオフセット位置から Floating point (float) の読み込み
OutReadSByte( <i>offset</i> )	出力データ領域のオフセット位置から Signed バイトの読み込み
OutReadUByte( <i>offset</i> )	出力データ領域のオフセット位置から Unsigned バイトの読み込み
OutReadSWord( <i>offset</i> )	出力データ領域のオフセット位置から Signed ワード (short) の読み込み
OutReadUWord( <i>offset</i> )	出力データ領域のオフセット位置から Unsigned ワード (short) の読み込み
OutReadSLong( <i>offset</i> )	出力データ領域のオフセット位置から Signed long ワード (long) の読み込み
OutReadULong( <i>offset</i> )	出力データ領域のオフセット位置から long ワード (long) の読み込み
OutReadString( <i>offset</i> )	出力データ領域のオフセット位置から readstring の読み込み
OutReadFloat( <i>offset</i> )	出力データ領域のオフセット位置から floating point value の読み込み

scanf()

注意：この機能は Email メッセージ内で使用できません。

この機能はストリング HTML フォーム内のオブジェクトをパスしたストリングを読み込み、フォーマット内の仕様によりストリングは解釈され、パスしたアーギュメントにより結果は出力データ領域に保存されます。ストリングのフォーマットは標準の C 機能呼び出し scanf() と同じです。

構文：

```
<?--#exec cmd_argument='scanf( "ObjName", "format", Arg1, ..., ArgN), ErrVal1, ..., ErrvalN'-->
```

- ObjName            - パスしたデータ・ストリングをもつオブジェクト名
- format            - パスしたストリングがフォーマットのされ方を指定します
- Arg1 - ArgN        - データの書き込み場所を指定します
- ErrVal1 -ErrValN   - 選択機能;エラーの場合に書き込む値/ストリングを指定します

文字	入力、アーギュメント・タイプ
d	十進数、バイト、ショート
i	番号、バイト、ショート。番号は八進数（先頭は 0（ゼロ））または十六進数（先頭は 0x または 0X）
o	八進数（先頭ゼロなし）、バイト、ショート
u	Unsigned 十進数；unsigned バイト、unsigned ショート
x	十六進数（先頭 0x または 0X）；バイト、ショート
c	文字；char*。次の入力文字（デフォルトでは 1）が表示された場所に配置されます。白いスペースを超えた標準のスキップは省略されます；次の白以外のスペースに文字を読み込むため %1s を使用します。
s	文字ストリング（引用されていない）；char*、ストリングと終端 "%0" のための十分大きな文字配列が追加されます。
e, f, g	オリジナルの数とフローティング - ポイント番号, 選択できる十進数ポイントと選択できるベキ指数があります；float*
%	文字上の %；アサインされません。

変換文字 d、i、o、u と x は l（文字エル）によって 'long' へのポインタが 'byte' または 'short' よりもアーギュメントリスト内に現れることを示します

SSI 機能 scanf へパスされたアーギュメントは：

アーギュメント	説明
OutWriteByte( <i>offset</i> )	出力データ領域のオフセット位置への byte の書き込み
OutWriteWord( <i>offset</i> )	出力データ領域のオフセット位置への word の書き込み
OutWriteLong( <i>offset</i> )	出力データ領域のオフセット位置への long の書き込み
OutWriteString( <i>offset</i> )	出力データ領域のオフセット位置への string の書き込み
OutWriteFloat( <i>offset</i> )	出力データ領域のオフセット位置への floating point value の書き込み

デフォルト出力：

```
Write succeeded
Write failed
```

## IncludeFile()

この機能は Web ページ上のファイルの内容を含みます。

### 構文:

```
<?--#exec cmd_argument='IncludeFile( "File name" )'-->
```

### デフォルト出力:

Success	- <File content>
Failure	- Failed to open <filename>

## SaveToFile()

**注意:** この機能は Email メッセージ内で使用できません。

この機能はパスしたフォームの内容をファイルへ保存します。パスした名前 / 値の対は "セパレータ " スtringで分割されたフィル " ファイル名 " に書き込まれます。[Append|Overwrite] パラメータは指定したファイルの上書きをするか、またはファイル内のデータが追加されるかを決めます。

### 構文:

```
<?--#exec cmd_argument='SaveToFile( "File name", "Separator",[Append|Overwrite] )'-->
```

### デフォルト出力:

Success	- Form saved to file
Failure	- Failed to save form

## SaveDataToFile()

**注意:** この機能は Email メッセージ内で使用できません。

この機能はパスしたフォームのデータをファイルへセーブします。“オブジェクト名”パラメータは選択機能で、指定される場合、そのオブジェクトからのデータのみが保存されます。指定されない場合は、フォーム内の全てのオブジェクトからのデータが保存されます。

### 構文:

```
<?--#exec cmd_argument='SaveDataToFile( "File name", "Object name",[Append|Overwrite] )'-->
```

### デフォルト出力:

Success	- Form saved to file
Failure	- Failed to save form

## SSI 出力の変更

SSI 機能からの出力ストリングを変更するには二つの方法があります。

1. システムの全ての SSI 機能のための出力ストリングを含む "¥ssi\_str.cfg" と呼ばれるファイルを作成することで SSI 出力デフォルトを変更
2. SSI 機能 "SsiOutput()" を呼び出すことによる SSI 出力の一時的な変更。

### SSI 出力ストリング・ファイル

ファイル・システム内にファイル "¥ssi\_str.cfg" が見つかりファイルが次の仕様に従っている場合、SSI 機能はデフォルト設定の代わりにファイル内で指定された出力ストリングを使用します。

ファイルは次のようにフォーマットされています：

```
[StoreEtnConfig]
Success: "String to use on success"
Invalid IP: "String to use when the IP address is invalid"
Invalid Subnet: "String to use when the Subnet mask is invalid"
Invalid Gateway: "String to use when the Gateway address is invalid"
Invalid Email server: "String to use when the SMTP address is invalid"
Invalid IP or Subnet: "String to use when the IP address and Subnet mask does not match"
Invalid DNS1: "String to use when the primary DNS cannot be found"
Invalid DNS2: "String to use when the secondary DNS cannot be found"
Save Error: "String to use when storage fails"
Invalid DHCP state: "String to use when the DHCP state is invalid"

[scanf]
Success: "String to use on success"
Failure: "String to use on failure"

[IncludeFile]
Failure: "String to use when failure"1

[SaveToFile]
Success: "String to use on success"
Failure: "String to use on failure"1

[SaveDataToFile]
Success: "String to use on success"
Failure: "String to use on failure"1

[GetText]
Success: "String to use on success"
Failure: "String to use on failure"
```

このファイルの内容は最初の 'File path' の行と次のファイルパスの行を置き換えることで書き換えることができます。

**例：**

```
[File path]
¥user¥ssi_strings.cfg

???????????????? 'user¥ssi_strings.cfg' ??????????
```

---

1. ‘%’ はストリングのファイル名を含みます。

## 一時的な SSI 出力変更

SSI 機能と呼ばれる次のための SSI 出力は SSI 機能 “SsiOutput()” と変更されます。SSI 機能  
を呼び出す次はこの呼び出しによって出力を使用します。

### 構文:

```
<?--#exec cmd_argument='SsiOutput( "Success string", "Failure string" )'-->
```

### 例:

この例は scanf SSI 呼び出しのための出力ストリングの変更の仕方を表示します。

```
<?--#exec cmd_argument='SsiOutput ( "Parameter1 updated", "Error" )'-->  
<?--#exec cmd_argument='scanf( "Parameter1", "%d", OutWriteByte(0) )'-->
```

# Email クライアント

## 概論

Email クライアントを組み込んで定義された入力 / 出力データ領域のトリガ・イベントに基づいた Email メッセージを送信することができます。クライアントは SSI をサポートしますが、SSI 機能には Email メッセージでは使用することができないものもあることに注意してください（各 SSI 機能毎によって異なります）。

- 8-1 “Server Side Include (SSI)” も参照してください。

### サーバの設定

ABC は Email メッセージを送信するため有効な SMTP サーバのコンフィグレーションが必要です。これらの設定はファイル・システム ‘`¥ethcfg.cfg`’ に保存されます。

- 4-2 “Ethernet コンフィグレーション・ファイル (‘`ethcfg.cfg`’)” も参照してください。

### イベント・トリガ・メッセージ

前述したように、Email クライアントは入力 / 出力データ領域に定義されたメッセージを送信できます。操作中は次のように動作します。

1. トリガ・ソースは指定された場所から呼び出されます。
2. 論理的 AND はトリガ・ソースとマスク値の間で動作します。
3. 結果はリファレンス値と比較されます。
4. 結果が True の場合、Email は指定された受信対象へ送信されます。

Which events that shall cause a particular message to be sent, is specified separately for each message. For more information, see 9-2 “Email 定義”.

Note that the Input- and Output Data areas are scanned twice per second, i.e. to ensure that an event is detected by the ABC, it must be present longer than 0.5 seconds.

# Email 定義

Email 定義は次の二つのディレクトリに保存されます：

- **‘%user%email’**  
このディレクトリは一般レベルの FTP ユーザにより変更可能な 10 までのメッセージを保存します。
- **‘%email’**  
このディレクトリはアドミニ・レベルの FTP ユーザにより変更可能な 10 までのメッセージを保存します。

Email 定義ファイルは ABC に適切に認識されるために ‘email\_1.cfg’, ‘email\_2.cfg’... ‘email\_10.cfg’ という名前が付けられなければなりません。

## ファイル・フォーマット：

```
[Register]
Area, Offset, Type

[Register Match]
Value, Mask, Operand

[To]
recipient

[From]
sender

[Subject]
subject line

[Headers]
Optional extra headers

[Message]
message body
```

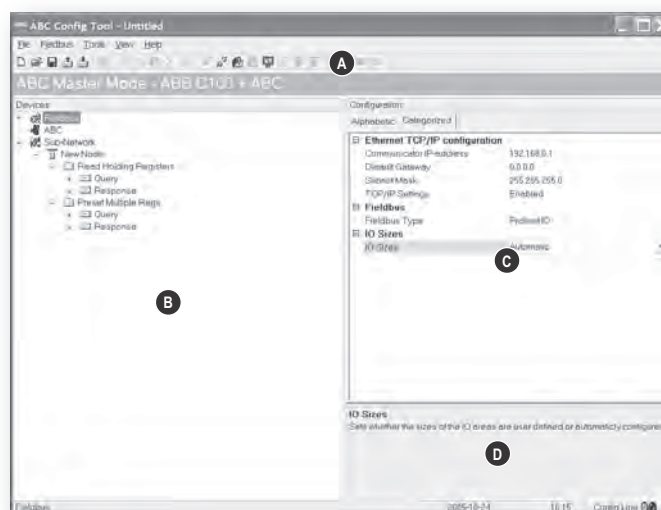
キー	値	SSI のための スキャン
Area	ソース領域。可能な値：‘IN’ ( 入力データ領域 ) あるいは ‘OUT’ ( 出力データ領域 )	いいえ
Offset	ソース・オフセット、十進数または十六進数で書き込み	
Type	ソース・データ・タイプ。可能な値は ‘バイト’、‘ワード’、‘ロング’	
Value	比較のためのレファレンス値として使用	
Mask	マスク値、比較用トリガ・ソースとして適用（論理的 AND）	
Operand	可能な値は ‘<’, ‘=’ あるいは ‘>’	
To	Email 受信対象	はい
From	送信側 Email アドレス	
Subject	Email 件名。1 行のみ	
Headers	選択機能；追加ヘッダの提供のために使用	
Message	実際のメッセージ	

注意：ABC によって扱われるため十六進数が接頭辞 ‘0x’ で書き込まれます。

# ABC コンフィグツールのナビゲーション

## メイン・ウィンドウ

ABC コンフィグツールのメイン・ウィンドウは次の4つのセクションに分けることができます：



- **A: プルダウン・メニューとツールバー**

左側の二番目のドロップダウン・メニューはそのときの選択で変わります。ツールバーによって最もよく使う機能に速くアクセスできます。

- **B: ナビゲーション・セクション**

このセクションはサブネットワーク・コンフィグレーションの異なるレベルを変更するメインツールです。

入力は '+' の表示か 'サブメニュー' で選ぶことができます。これらのパラメータにアクセスするには、 '+' をクリックしエントリを展開する必要があります。

ナビゲーション・ウィンドウには Fieldbus、ABC、Sub-network という名前の3つのメインレベルがあります。このセクションのエントリを右クリックすることで、さらに特別なエントリに関連した選択が表示されます。

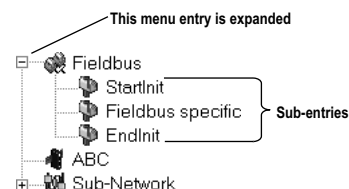
- **C: パラメータ・セクション**

このセクションはパラメータのリストまたは現在ナビゲーション・セクションで選択しているエントリに関連した選択肢が含まれます。パラメータの値は、パラメータごとにセクション・ボックスまたは手動で特定（設定）することができます。

値は 10 進数（例えば '42'）、または 16 進数（例えば '0x2A'）フォーマットで特定できます。

- **D: 情報セクション**

このセクションは現在選択されているパラメータに関連した情報が含まれます。



Configuration:	
Alphabetic	Categorized
Communication	
Brate (bits/s)	9600
Data bits	8
Parity	None
Physical standard	RS232
Start bits	1
Stop bits	1
Timing	
Message delimiter (10ms)	0

Parameter Section

Message delimiter (10ms)
The time between transaction

Information Section

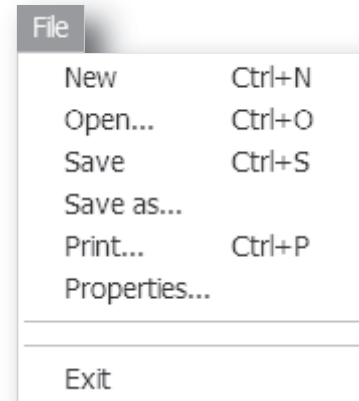


## プルダウン・メニュー

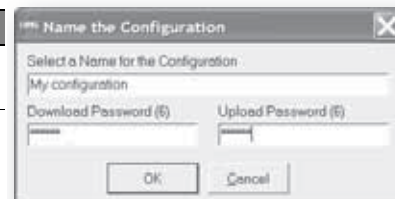
### File

このメニューには次のエントリがあります：

- **New**  
新しいコンフィグレーションの作成  
12-1 “コンフィグレーション・ウィザード” もご参照ください。
- **Open...**  
既に作成されたコンフィグレーションを開きます。
- **Save**  
現在のコンフィグレーションを保存します。
- **Save As...**  
新しい名前で現在のコンフィグレーションを保存します。
- **Print...**  
現在のコンフィグレーションの詳細を印刷します。
- **Properties...**  
次のウィンドウを表示します：



項目	説明
Select a Name for the Configuration	コンフィグレーション名を入力します。
Download Password(6)	これらのフィールドはコンフィグレーションをパスワード・プロテクトする場合に使用します。
Upload Password(6)	



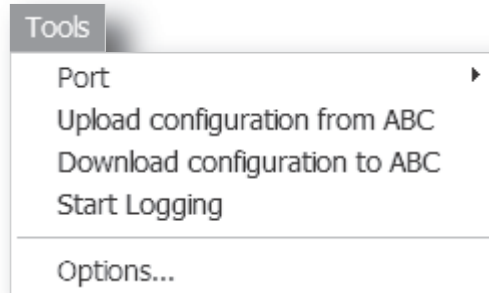
**注意：**安全な場所にパスワードのコピーを保管してください。パスワードを失くした場合データは使用できません。

- **Exit**  
ABC コンフィグツールを終了します。

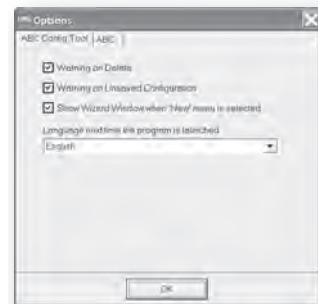
## Tools

このメニューには次のエントリがあります：

- Port**  
 この入力はゲートウェイのコンフィグレーションに使用する COM ポートを選択します。
- Upload configuration from ABC**  
 ゲートウェイから ABC コンフィグツールへコンフィグレーションをアップロードします。
- Download configuration to ABC**  
 ゲートウェイへコンフィグレーションをダウンロードします。
- Start Logging**  
 データ・ロガーを開始します（18-1 “データ・ロガー”をご参照ください）  
 データ・ロガーが動作しているとき、このメニューは ‘Stop Logging’ に変わります。
- Options**  
 このメニューは次のウィンドウを表示します。



項目	説明
Warning on Delete	何か削除される毎に確認のダイアログが表示されます。
Warning on unsaved data	セーブされていない ABC コンフィグツールのデータがある場合、確認のダイアログが表示されます。
Show Wizard when “New” menu is selected	新しいコンフィグレーションが作成されるとウィザードが表示されます。
Language next time the program is launched	使用する言語を選択します。新しい設定は次にプログラムが開始されると有効になります。



ABC タブを選択するとさらにプロパティが現れます。

項目	説明
Size of logbuffer	データ・ロガー はデフォルトで各方向で 512 エントリまでログをとることができます。もし必要であれば、異なる数のエントリを指定することができます（有効な設定範囲は 1...512）。 ‘Apply’ をクリックし新しい設定を有効にしてください。18-1 “データ・ロガー” もご参照ください。
Firmware Download	組み込みフィールドバス・インターフェースにファームウェアのダウンロードをします。 <b>警告：ご使用には注意してください。</b>



項目	説明
Factory Restore	ゲートウェイのファームウェアを出荷時の状態に戻します（組み込みフィールドバス・インターフェースには影響しません）。
Block Configuration	ゲートウェイによってダウンロード・コンフィグレーションは実行されません。 <b>警告：ご使用には注意してください。</b>
Create Error log	エラー・ログファイルを作成します。

## View

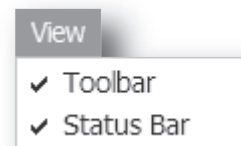
このメニューには次のエントリがあります：

- **Toolbar**

このエントリはメイン・ウィンドウのツールバー・アイコンを有効 / 無効にします。

- **Status Bar**

このエントリはメイン・ウィンドウのステータスバーを有効 / 無効にします。



## Help

このメニューには次のエントリがあります：

- **Contents**

オンライン・ヘルプ・システムの目次を表示します。

**注意：**書き込み時、オンライン・ヘルプ・システムは表示されません。

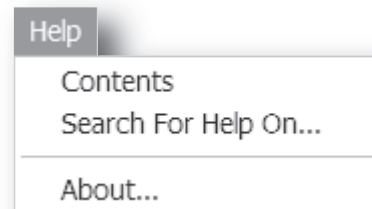
- **Search For Help On...**

オンラインヘルプで特定の内容を探します。

**注意：**書き込み時、オンライン・ヘルプ・システムは表示されません。

- **About...**

ゲートウェイと現在の ABC コンフィグツールのバージョンについての一般的な情報を表示します。



## ツールバー・アイコン

ツールバーは最もよく使用される機能のアイコンの集まりです。

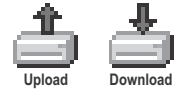
- **New, Open & Save**

10-2 “File” を参照してください。



- **Upload from ABC & Download to ABC**

10-3 “Tools” を参照してください。



- **Up one Level**

このアイコンをクリックし、ナビゲーション・セクションの中の選択を移動します。



- **Cut, Copy, Paste, Delete, Insert**

これらのアイコンはナビゲーション・セクションの中の共通の編集機能のために使用されます。



- **Connect**

このアイコンをクリックし、ABC コンフィグツールをゲートウェイに接続します。



- **Disconnect**

このアイコンをクリックし、ABC コンフィグツールをゲートウェイから切断します。



- **Start Logging & Stop Logging**

10-3 “Tools” と 18-1 “データ・ロガー” を参照してください。



- **Sub-Network Monitor**

このアイコンをクリックし、Sub-network Monitor を起動します。  
(16-1 “サブネットワーク・モニタ” を参照してください)。



- **Add Command**

このアイコンは現在選択されているノードにコマンドを追加するために使用されます。



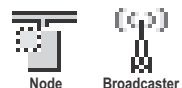
- **Add Mailbox**

(メールボックス・メッセージ機能については 22-1 “メールボックス・エディタ” を参照してください)



- **Add Node & Add Broadcaster**

これらのアイコンはコンフィグレーションにノードを追加するために使用されます。



- **Node Monitor**

このアイコンをクリックすることで、Node Monitor を起動します。(17-1 “ノード・モニタ” を参照してください)



- **Add Transaction(s)**

これらのアイコンは選択されたノードにトランザクションを追加するために使用されます。



## 基本設定

### フィールドバス設定

(このセクションに記載されているパラメータにアクセスするためにナビゲーション・セクションの‘フィールドバス’を選択してください)。



#### 概論

スタートアップ中 ABC のフィールドバス・インターフェースは ABC コンフィグツールで生成されたコンフィグレーションに合わせて初期化されます。オプションでいくつかのパラメータをマニュアルで ABC でのデータ処理をより制御しやすいように設定することができます。

#### Fieldbus Settings

To be able to participate on the network, the following settings must be set properly:

- **Fieldbus Type**

ABC コンフィグツールは幅広いネットワーク・システムをサポートしています。このパラメータが ‘DeviceNet’ に設定されていることを確認してください。



Fieldbus Type

- **Modbus Address Mode**

Enabled- Use Modbus Address Mode

Disabled- Use Anybus Address Mode

See also 2-7 “Modbus/TCP”

- **Communicator IP-address, Gateway, Subnet Mask**

See 4-1 “基本的なネットワーク・コンフィグレーション”

- **TCP/IP Settings**

Enabled- Use settings in ABC Config Tool

Disabled- Use settings stored in ‘ethcfg.cfg’

See also 4-1 “基本的なネットワーク・コンフィグレーション”



IO Sizes

#### IO Sizes

これらのパラメータは内部メモリ・バッファからのデータがどのように交換されるのかを指定します。これはサブネットワーク・コンフィグレーション上で自動的に処理されるか、または手動で指定されます。

- **Automatic**

全てのデータは DeviceNet 上の I/O データとして表示されます。

- **User defined**

追加のパラメータ・プロパティ、‘IO Size In’ と ‘IO Size Out’ が表示されます。指定された合計はメモリ・バッファのアドレス 0x0000 から始まり予約され、I/O データとして表示されます。残りはパラメータ・データのために予約されます。

2-6 “EtherNet/IP” も参照してください。

## ABC パラメータ

(このセクションに記載されているパラメータにアクセスするためには Navigation セクションの「ABC」を選択してください。)



### Interface

現在はシリアル通信のみサポートしています。

### Status / Control Word

(20-1 “制御 / ステータス・レジスタ” を参照してください)。.

値	説明
Enabled	制御 / ステータス・レジスタを有効にします。 コントロールレジスタの「データ有効」ビットはサブ・ネットワーク通信の開始のために設定される必要があります。
Enabled but no startup lock	制御システムがサブネットワーク通信の開始をするために「データ有効」ビットの設定を必要としないこと以外はこの設定は「有効」に類似した設定です。
Disabled	この設定は制御 / ステータス・レジスタを完全に無効にします。

### Module Reset

このパラメータはエラー発生時のゲートウェイの動作を特定します。

値	説明
Enabled	ゲートウェイは再起動され、エラーは表示されません。
Disabled	ゲートウェイは停止し、エラーが表示されます。

### Protocol Mode

このパラメータはサブネットワークを使用するためのモードを特定します。

値	説明
Generic Data Mode	このモードはゲートウェイとサブネットワーク上のノード間のマスター / スレーブがあるプロデュース & コンsum・ベースのプロトコルを意図しています。
Master Mode	このモードは1つのマスターが多数のスレーブとデータ交換する「クエリ & レスポンス」ベースのプロトコルを意図しています。

2-4 “プロトコル・モード” も参照してください。

### Statistics

この伝送 / 受信カウンタはサブネットワーク上でトランザクションがどれくらい交換されたかを表示します。この機能はデバッグを目的としています。

- **Receive Counter Location**  
内部メモリ・バッファ内の受信カウンタの位置を特定します。
- **Transmit Counter Location**  
内部メモリ・バッファ内の伝送カウンタの位置を特定します。

両カウンタとも「スタティスティクス」から「有効」へ設定することで有効になります。

## サブネットワーク・パラメータ

(このセクションに記述されているパラメータにアクセスするには、Navigation セクションで ‘Sub Network’ を選択してください。)



### Communication

このパラメータはサブネットワーク向けに使用される実際の通信の設定を特定します。

パラメータ	説明	有効設定
Bit rate	ビットレートの選択	1200...57600
Data bits	データビット数の選択	7, 8
Parity	パリティ・モードの選択	None, Odd, Even
Physical standard	物理インターフェースタイプの選択	RS232, RS422, RS485
Start bits	スタートビット数	1
Stop bits	ストップビット数	1, 2

### Start- and End Character

**注意：** これらのパラメータはジェネリック・データ・モードでのみ使用可能です。

スタート / エンド・キャラクタはシリアル・メッセージの始まりと終わりを示します。例えば、メッセージは <ESC> とともに表示され、<LF> で終了します。この場合、スタート・キャラクタは 0x1B (<ESC> の ASCII コード) になり、エンド・キャラクタは 0x0A (<LF> の ASCII コード) になります。

パラメータ	説明	有効設定
End Character Value	メッセージのためのエンド・キャラクタ、ASCII	0x00 - 0xFF
Use End Character	エンド・キャラクタの使用の決定	Enable / Disable
Start Character Value	メッセージのためのスタート・キャラクタ、ASCII	0x00 - 0xFF
Use Start Character	スタート・キャラクタの使用の決定	Enable / Disable

### Timing (Message Delimiter)

このカテゴリのパラメータはプロトコル・モードによって僅かに異なります。

- マスター・モード

メッセージ・デリミタは 10ms 毎に二つのメッセージを分割する時間を特定します。0（ゼロ）に設定するとゲートウェイは 3.5 キャラクタの標準 Modbus デリミタを使用します（実際の ms 数はその時使用している通信の設定に基づき自動的に計算されます）。

- ジェネリック・データ・モード

メッセージ・デリミタは 10μs 単位で二つのメッセージを分割する時間を特定します。



# ノード

## 概論

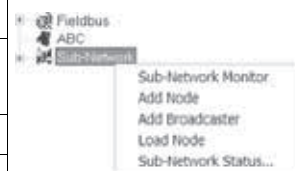
ABC コンフィグツールでは、ノードはネットワーク上でシングル・デバイスとして機能します。ゲートウェイは伝統的な意味でのスキャンリストを使用せず、すべてのノードとそのトランザクションの処理は ABC コンフィグツールで設定された順番に実行されます。

ABC コンフィグツールで作成されるノードの最大数は 31 です。

## ノードの追加と管理

(これらの機能にアクセスするためには Navigation セクション内の ‘Sub Network’ 上で右ボタンをクリックしてください)

機能	説明
Paste	クリップボードからノードを貼り付けます
Sub Network Monitor	サブネット・モニタを起動します (16-1 “サブネットワーク・モニタ”)
Add Node	コンフィグレーションにノードを追加します
Add Broadcaster <sup>a</sup>	コンフィグレーションにブロードキャスタ・ノードを追加します
Load Node	前にセーブされたノードを追加します
Sub-Network Status...	サブネットワークについての診断情報を表示します



a. この機能はマスター・モードのみ使用できます。

## ノード・パラメータ

(このセクションに記述されるパラメータへアクセスするためには Navigation セクション内のノードを選択してください)。

パラメータ	説明
Slave Address	ここで入力される値は特定コマンド内のノード・アドレスの設定に使用されます。 詳細は 15-3 “コマンド・エディタ” を参照してください。



# トランザクション

## 概論

前述した通り、トランザクションは実際のシリアル・サブネットワーク上でデータ交換されるシリアル・テレグラムを表します。ゲートウェイは従来の意味でのスキャンリストを使用しないため、全てのノードとそのトランザクションは ABC コンフィグツールで定義された順番で処理されます。

トランザクションは2つのわずかに異なるプロトコル・モードで処理されます。

- マスター・モード

一般のノード向けにトランザクションはクエリと応答と常に対になっています。クエリはゲートウェイによって発行され、応答はサブネットワーク上のスレーブにより発行されます。ブロードキャストはトランザクションのみ送信することができます。

- ジェネリック・データ・モード

トランザクションは双方向に追加することができます。サブネットワークへ送信されたトランザクションは 'Transaction Produce' と呼ばれ、他のノードにより発行されたトランザクションは 'Transaction Consume' と呼ばれます。

理論的にはゲートウェイは 100 トランザクションまでサポートしています。実際の数としてはそれよりも少なく、トランザクションが設定されたメモリの制限に依存します。

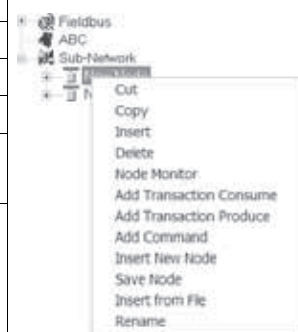
## トランザクションの追加と管理

(これらの機能にアクセスするためには Navigation セクションでマウスを右クリックしてください)

機能	説明
Cut	ノードを切り取りクリップボードへ保存します
Copy	ノードをクリップボードへコピーします
Insert	クリップボードからノードを挿入します
Delete	ノードを削除します
Node Monitor	ノード・モニタの起動をします (17-1 "ノード・モニタ")
Add Transaction(s) <sup>a</sup>	一般ノード上でクエリと応答を追加します。可読性を増すためにふたつのトランザクションはグループ化されます。ブロードキャスト上では、ひとつのトランザクションが追加されます。
Add Transaction Consume <sup>b</sup>	'Consume' トランザクションを追加します
Add transaction Produce <sup>b</sup>	'Produce' トランザクションを追加します
Add Command	事前に設定されたトランザクションをノードに追加します
Insert New Node	現在選択されたノードに新しいノードを挿入します
Save Node	選択されたノードを保存します
Insert from File	現在選択されたノードの上に事前に保存されたノードを挿入します
Rename	可読性を増すために各ノードにはこの機能で固有の名前が付けられます

a. マスター・モードのみ利用可能

b. ジェネリック・データ・モードのみ利用可能



## トランザクション・パラメータ（マスター・モード）

### パラメータ（クエリとブロードキャスト）

（これらのパラメータにアクセスするためには Navigation セクションでクエリまたはブロードキャスト・トランザクションを選択してください）

パラメータ	説明
Minimum time between broadcasts (10ms)	このパラメータはゲートウェイがブロードキャスト・トランザクションを送信後、スキャンリストで次のエントリ処理を行うまでの時間を指定します。スレーブ機器がブロードキャストの処理を終えるために十分大きな値を設定する必要があります。単位はミリセカンド（ms）で入力され、値は 10 の倍数、最小値は 10ms です。 <b>注意：</b> この設定はブロードキャスト・ノードのみに適応します。
Offline options for field-bus	このパラメータは上位ネットワークがオフラインになったときの動作を指定します。この設定はサブネットワークへ送信されるデータに影響を与えます。 <ul style="list-style-type: none"> <li>• Clear - スレーブ機器へのデータがクリアされます（ゼロに設定されます）</li> <li>• Freeze - スレーブ機器へのデータがフリーズされます</li> <li>• NoScanning - サブネットワークの更新が停止します</li> </ul>
Offline options for sub-network	このパラメータはサブネットワークがオフラインになったときの動作を指定します。この設定は制御システムへ返信されるデータに影響を与えます。 <ul style="list-style-type: none"> <li>• Clear - サブネットワークがオフラインになったとき上位ネットワークのデータがクリア（0）されます。</li> <li>• Freeze - サブネットワークがオフラインになったとき上位ネットワークのデータがフリーズされます。</li> </ul>
Reconnect time (10ms)	このパラメータはゲートウェイが切断されたノードに再接続を開始するまでの時間を指定します。最大リトライ（下記参照してください）回数に達した場合ノードは切断されます。単位はミリセカンド（ms）で入力され、値は 10 の倍数、最小値は 10ms です。 <b>注意：</b> この設定はブロードキャスト・ノードのみに適応します。
Retries	このパラメータはノードが切断されるまでに連続して発生するタイムアウトの回数を指定します。
Timeout time (10ms)	このパラメータはノードからの応答の待機時間を指定します。この時間を超える場合、ゲートウェイは最大リトライ回数（上記を参照してください）までクエリを再送します。単位はミリセカンド（ms）で入力され、値は 10 の倍数、最小値は 10ms です。
Trigger byte address	このパラメータは内部メモリのトリガ・バイトの場所を指定します（'Update mode' が 'Change of state on trigger' に設定されているときのみ適応）
Update mode	このパラメータはスレーブにトランザクションが送信される時を指定します。 <ul style="list-style-type: none"> <li>• Cyclically 'Update time' パラメータで指定された時間間隔で周期的にトランザクションを送信します。</li> <li>• On data change データ領域は更新時間で設定された時間間隔でポーリングされます。データの変更が検出されるとトランザクションは送信されます。</li> <li>• Single shot スタート時にクエリが送信されます。</li> <li>• Change of state on trigger トリガ・バイト値が変更されるとクエリが送信されます。この機能は制御システムがゲートウェイに特殊なクエリの送信を知らせることを可能にします。この機能を正確に使用するために、制御システムはクエリ/トランザクションに関連するデータ領域を速く更新し、トリガ・バイトを一つずつ増加させる必要があります。トリガ・バイトの場所は 'トリガ・バイト・アドレス' パラメータによって指定されます。</li> </ul>
Update time (10ms)	このパラメータはトランザクションが 10ms 単位で送信される頻度を指定します（'更新モード' が 'Cyclically' に設定される場合のみ適応）

## パラメータ（レスポンス）

（これらのパラメータへアクセスするためには Navigation セクションの応答トランザクションを選択してください）

パラメータ	説明
Trigger byte	このパラメータは応答のためのトリガ機能を有効/無効に切り替えるために使用されます。有効に設定されている場合、ゲートウェイはサブネットワークから新しいデータを受信するごとに、トリガ・バイトを 1 ずつ増加させます。この機能は制御システムにデータの更新を知らせるために使用されます。 トリガ・バイトの位置は下の「トリガ・バイト・アドレス」パラメータにより指定されます。
Trigger byte address	このパラメータは内部メモリ・バッファのトリガ・バイトの位置を指定します。 有効な設定範囲は 0x000... 0x1FF と 0x400... 0xNNN です。

## トランザクション・パラメータ（ジェネリック・データ・モード）

### プロデュース・トランザクション (Produce- Transactions)

（これらのパラメータにアクセスするためには Navigation セクションのプロデュース・トランザクションを選択してください）

パラメータ	説明
Offline options for fieldbus	このパラメータは上位ネットワークがオフラインになった場合、このトランザクションのために取られる動作を指定します。この設定はサブネットワークへ送信されるデータに影響を与えます。 <ul style="list-style-type: none"> <li>• Clear 上位ネットワークがオフラインになったときサブネットワーク上のデータはクリア (0) されます。</li> <li>• Freeze 上位ネットワークがオフラインになったときサブネットワーク上のデータはフリーズされます。</li> <li>• NoScanning 上位ネットワークがオフラインになったときサブネットワーク上でのこのトランザクションのスキャンを停止します。</li> </ul>
Update mode	トランザクションの更新モード： <ul style="list-style-type: none"> <li>• Cyclically トランザクションは「更新時間」パラメータで指定された時間間隔で周期的に送信されます。</li> <li>• On data change データ領域は変更確認のために更新時間パラメータで設定された時間間隔でポーリングされます。データ内で変更が検出されるとトランザクションが送信されます。</li> <li>• Single shot スタート時にトランザクションは一度送信されます。</li> <li>• Change of state on trigger トリガ・バイトが変更されるとトランザクションが送信されます。この機能により制御システムがゲートウェイに特殊なトランザクションを送信することを通知することができます。この機能を正しく使用するために、制御システムはまずこのトランザクションに関連したデータ領域を更新し、1 ずつトリガ・バイトを増加させる必要があります。トリガ・バイトの位置は「トリガ・バイト・アドレス」パラメータで指定されます。</li> </ul>

パラメータ	説明
Update time (10ms)	このパラメータはトランザクションが 10ms 単位で送信される頻度を指定します（'更新モード' が 'Cyclically' に設定される場合のみ適応）。
Trigger byte address	<p>このパラメータは内部メモリ・バッファ内のトリガ・バイトの位置を指定します。'Update mode' が 'Change of state on trigger' に設定されるとこのパラメータで指定されたメモリの位置はゲートウェイでモニタされます。トリガ・バイトが更新されるとゲートウェイはサブネットワーク上にトランザクションを作成します。</p> <p>この方法是对应するトリガ・バイトを更新することで制御システムがゲートウェイにサブネットワーク上に固有のトランザクションを作成させることができます。</p> <p>トリガ・バイトは動作毎に1ずつ増加します。トリガ・バイト・アドレスは各トランザクションに対して一つである必要があることに注意してください。</p> <p><b>注意：</b>'Update mode' パラメータが 'Change of state on trigger' に設定されない限りこのパラメータは無効です。</p>

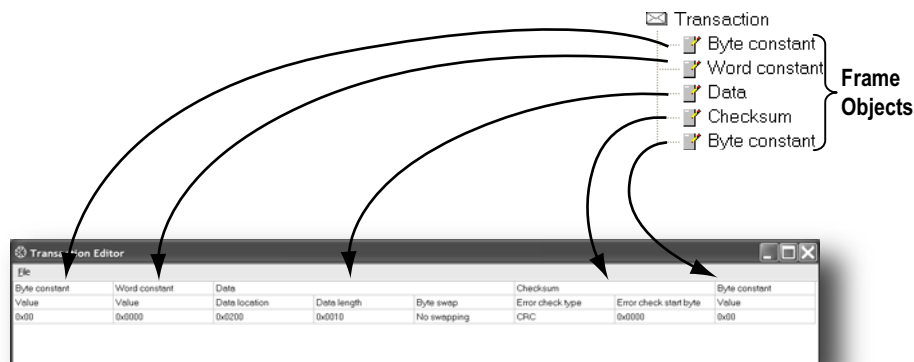
## コンスーム・トランザクション (Consume- Transactions)

(これらのパラメータにアクセスするためには Navigation セクションからコンスーム・トランザクションを選択してください)

パラメータ	説明
Offline options for sub-network	<p>このパラメータはサブネットワークがオフライン時のこのトランザクションのための動作を指定します。</p> <p>このパラメータは上位ネットワークに送信されるデータに影響します。</p> <ul style="list-style-type: none"> <li>• Clear サブネットワークがオフライン時上位ネットワーク上のデータはクリアされます (0)</li> <li>• Freeze サブネットワークがオフライン時上位ネットワークのデータはフリーズします</li> </ul>
Offline timeout time (10ms)	このパラメータは 2 つの入力メッセージの間の時間の最大値を 10ms 単位で指定します。この時間を超えるとサブネットワークはオフラインとみなされます。0 の値を設定するとこの機能は無効になり、サブネットワークはオフラインになりません。
Trigger byte	<ul style="list-style-type: none"> <li>• Enable トリガ・バイトを有効にします。トリガ・バイトの位置は 'Trigger byte address'（下記参照してください）で指定されなければなりません。トリガ・バイト値はゲートウェイによって有効なトランザクションがコンスームされる毎に増加します。この機能はサブネットワーク上で新しいデータがコンスームされる毎に制御システムに通知することを可能にします。</li> <li>• Disable トリガ・バイト機能を無効にします。</li> </ul>
Trigger byte address	<p>このパラメータは内部メモリ・バッファのトリガ・バイトの位置を指定します。</p> <p>有効な設定範囲は 0x000... 0x1FF と 0x400... 0xNNN です。</p> <p>トリガ・バイト・アドレスは各トランザクションに対して一つである必要があることに注意してください。二つ以上のトランザクションとは配分できません。</p>

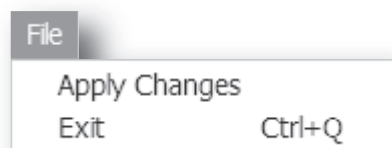
## トランザクション・エディタ

トランザクション・エディタはトランザクションの個々のフレーム・オブジェクトを編集するために使用することができます。同じ設定はまたメイン・ウィンドウの Parameter セクションでも可能ですが、トランザクション・エディタはより視覚的方法でフレーム・オブジェクトを表示します。



パラメータの値の編集をするためクリックしキーボードを使用して新しい値を入力してください。既に設定されたコマンドを使用しトランザクションの編集をしているとき、トランザクションによっては編集できないかもしれません。

File-メニューには次のエントリがあります：



- **Apply Changes**  
変更をすべて保存して終了しメイン・ウィンドウに戻ります。
- **Exit**  
保存しないで終了します。

例：



この例で生成されたトランザクションは次のような構成になります：

最初のバイトはデータ・フィールド長（この場合 8）を指定する 2 バイトを伴う STX (0x02) を保持します。次の 8 バイトはデータでこれは、クエリ・トランザクションであるためデータはアドレス位置 0x0202 で始まる出力領域から取り込まれます。データ上ではスワップされません。これは 2 バイト・チェックサムに従います。チェックサム計算はトランザクションの 2 番目のバイトで始まります。

トランザクションは常に ETX (0x03) で終了します。

## フレーム・オブジェクト

### 概論

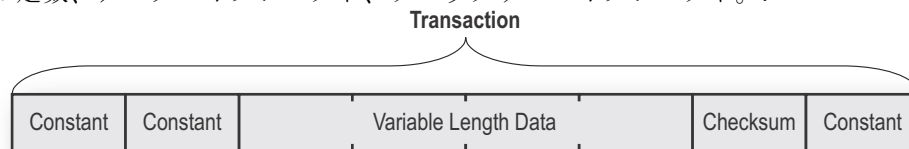
各トランザクションはシリアル・テレグラム・フレームを生成するフレーム・オブジェクトを含みます。各フレーム・オブジェクトはゲートウェイにテレグラムの特定の部分の処理方法または生成方法を指定します。

この章では5種類のフレーム・オブジェクトについて説明します：

- 通常オブジェクト
- 制限オブジェクト
- データ・オブジェクト
- 変数データ・オブジェクト
- チェックサム・オブジェクト

例：

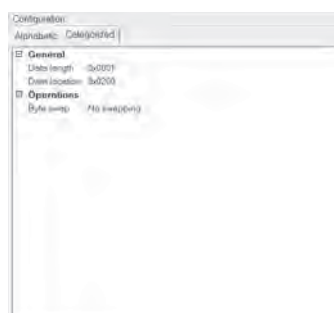
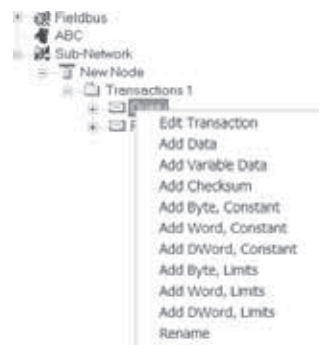
次のトランザクションは複数のフレーム・オブジェクトで構成されています；三つの定数、データ・オブジェクト、チェックサム・オブジェクト。.



### フレーム・オブジェクトの追加と編集

トランザクションにフレーム・オブジェクトを追加するため、Navigation セクション内のトランザクション上で右クリックし、表示されるメニューの中からエントリを一つ選択します。

“Transaction Editor”と呼ばれるエントリが、トランザクションとフレーム・オブジェクトをより視覚的に編集するために使用されるトランザクション・エディタを起動します。詳細については 13-5 “トランザクション・エディタ”を参照してください。



Data Object, Parameters

特定のフレーム・オブジェクトに関連するパラメータを編集するため、Navigation セクション内のフレーム・オブジェクトを選択してください。フレーム・オブジェクトの設定は Parameter セクション内に表示されます。

トランザクション・エディタを使用してより視覚的にトランザクション内のフレーム・オブジェクトの編集が可能です。13-5 “トランザクション・エディタ”を参照してください。

## 通常オブジェクト (バイト、ワード、ダブルワード)

通常オブジェクトには定数と 3 種類の大きさがあります。

- バイト  
8 ビット
- ワード  
16 ビット
- ダブルワード  
32 ビット

トランザクションの方向により通常オブジェクトは異なる処理がされます。

- プロデュース / クエリ・トランザクション  
ゲートウェイは値が処理されていないときに値を送信します。
- コンスーム / レスポンス・トランザクション  
ゲートウェイは受信したバイト / ワード / ダブルワードと指定した値の一致を確認します。一致していない場合にはメッセージは破棄されます。

オブジェクトの値を設定するため、Navigation セクション内でこのメニューを選択し、Parameter セクション内で設定する値を入力します。

パラメータ	説明
Value	通常値



## 制限オブジェクト (バイト、ワード、ダブルワード)

制限オブジェクトには決まった値の範囲と 3 つの大きさがあります：

- バイト  
8 ビット
- ワード  
16 ビット
- ダブルワード  
32 ビット

トランザクションの方向によって制限オブジェクトは異なる処理がされます：

- プロデュース / クエリ・トランザクション  
このオブジェクトはこのようなトランザクションのために使用されません（値は定義されません）
- コンスーム / レスポンス・トランザクション  
ゲートウェイは受信したバイト / ワード / ダブルワードが指定した範囲以内であることを確認します。指定した範囲以内でない場合にはメッセージは破棄されます。

3 種類のインターバル・オブジェクトがあります：

- バイト  
8 ビット・インターバル
- ワード  
16 ビット・インターバル
- ダブルワード  
32 ビット・インターバル

オブジェクトの範囲を設定するため、Navigation セクション内でこれを選択し Parameter セクション内で次のように範囲を入力します。

**注意：** 値は最大値よりも小さくなければなりません（下記参照してください）

パラメータ	説明
Maximum Value	<p>これは範囲として設定できる値の最大値です。</p> <p>範囲： 0x00... 0xFFh                    (バイト)</p> <p>         0x0000... 0xFFFFh            (ワード)</p> <p>         0x00000000... 0xFFFFFFFFh   (ダブルワード)</p> <p><b>注意：</b> 値は最小値よりも大きくなければなりません（下記参照してください）</p>
Minimum Value	<p>これは範囲として設定できる値の最小値です。</p> <p>範囲： 0x00... 0xFEh                    (バイト)</p> <p>         0x0000... 0xFFFEh            (ワード)</p> <p>         0x00000000... 0xFFFFFFFh   (ダブルワード)</p> <p><b>注意：</b> 値は最大値よりも小さくなければなりません（上記参照してください）</p>

## データ・オブジェクト

データ・オブジェクトは次のように未加工のデータを表示するために使用されます：

- **プロデュース / クエリ・トランザクション**  
指定されたデータ・ブロックは上位ネットワークからサブネットワークへ転送されます。
- **コンスーム / レスポンス・トランザクション**  
指定されたデータ・ブロックはサブネットワークから上位ネットワークへ転送されます。

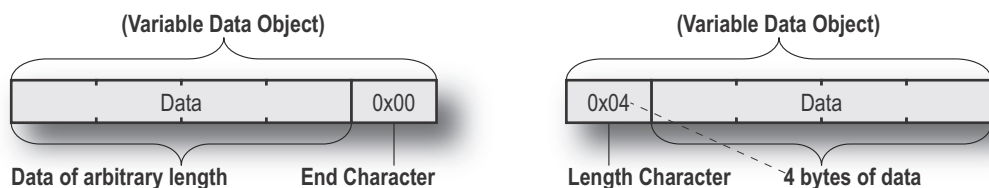
オブジェクトのプロパティを指定するため、Navigation セクション内でこれを選択し、次のように Parameter セクションで設定する値を入力します：

パラメータ	説明
Byte Swapping	<ul style="list-style-type: none"> <li>• No Swapping データのスワップはされません</li> <li>• Swap 2 bytes A, B, C, D が B, A, D, C になります</li> <li>• Swap 4 bytes A, B, C, D が D, C, B, A になります</li> </ul>
Data Length	バイト単位でのデータ・ブロックの長さです。レスポンスまたはコンスーム・トランザクションの場合、ここで指定する値と異なるデータサイズの入力メッセージは破棄されます。1 フレームの最大データ長は 300 バイトです。
Data Location	内部メモリ・バッファのデータ・ブロックの位置

## 変数データ・オブジェクト

**注意：** 各トランザクションに対して 1 つの変数データ・オブジェクトの送信ができます。

データ長が予め設定されていないことを除けばこのオブジェクトはデータ・オブジェクトと似ています。代わりに終端またはデータ長キャラクタが次のようにデータ・ブロック・サイズを指定します。



- **プロデュース / クエリ・トランザクション**  
指定されたデータ・ブロックは上位ネットワークからサブネットワークへ転送されます。  
ゲートウェイがデータ・ブロックの大きさを知るため制御システムは終端または長キャラクタを与える必要があります。終端またはデータ長キャラクタはそれ自体どちらか一方がサブネットワークへ転送されるか破棄されます。

• コンスーム / レスポンス・トランザクション

指定されたデータ・ブロックはサブネットワークから上位ネットワークへ転送されます。終端または長キャラクタはゲートウェイによって自動的に作成されます。(適用可能な場合) 終端またはデータ長キャラクタのどちらかは上位ネットワークへ転送されるか破棄されます。

オブジェクトのプロパティの指定をするため Navigation セクション内で変更するパラメータを選択し Parameter セクションに次のように設定を入力します：

パラメータ	説明
Byte Swapping	<ul style="list-style-type: none"> <li>• No Swapping データのスワップはされません</li> <li>• Swap 2 bytes A, B, C, D が B, A, D, C になります</li> <li>• Swap 4 bytes A, B, C, D が D, C, B, A になります</li> </ul>
Fill unused bytes	<ul style="list-style-type: none"> <li>• Enabled<sup>a</sup> ‘フィルタ・バイト’内で指定した値で未使用のデータを埋めます。</li> <li>• Disabled 未使用のデータを値で埋めません</li> </ul>
Filler byte	フィルタ・バイト値。‘未使用バイトの記入’が有効の場合のみ使用されます。
Data Location	データが呼び出し / 書き込みされる内部メモリ・バッファのオフセット
Object Delimiter	<ul style="list-style-type: none"> <li>• Length Character Length Character は内部メモリ・バッファで見ることができますが、サブネットワークでは見ることはできません。</li> <li>• Length Character Visible 長キャラクタは内部メモリ・バッファとサブネットワークの両方で見ることができます。</li> <li>• End Character 終端キャラクタは内部メモリ・バッファで見ることができますが、サブネットワークでは見ることはできません。</li> <li>• End Character Visible 終端キャラクタは内部メモリ・バッファとサブネットワークの両方で見ることができます。</li> <li>• No Character<sup>a</sup> 内部メモリ・バッファで終端または長キャラクタは生成されません。</li> </ul>
End Character Value	終端キャラクタ値 <sup>b</sup>
Maximum Data Length	設定可能な変数データ・オブジェクトのデータ長 (バイト単位)。実際のデータ長がこの値を超えた場合、メッセージは破棄されます。1 フレームの最大データ長は 300 バイトです。

a. コンスーム / レスポンス・トランザクションのみに適応されます

b. ‘オブジェクト・デリミタ’が‘終端キャラクタ’または‘終端キャラクタ可視’に設定されているときのみ使用します。

## チェックサム・オブジェクト

ほとんどのシリアル・プロトコルは転送中にデータの破損の検証をします。チェックサム・オブジェクトはトランザクション内のチェックサムを計算しその結果を含んでいます。

パラメータ	説明
Error Check Start byte	このパラメータはトランザクション内でチェックサム計算を開始するバイト・オフセットを指定します
Error Check Type	<p>このパラメータは使用するアルゴリズムの種類を指定します：</p> <ul style="list-style-type: none"><li>• CRC (2 バイト) 0xFFFF による CRC-16 多項式 (Modbus RTU 標準)</li><li>• LRC (1 バイト) unsigned 8 ビット値として全てのバイトが足されます。二番目の余数の結果がチェックサムに使用されます。</li><li>• XOR (1 バイト) 全てのバイトは理論的には XOR されます。結果はチェックサムとして使用されます。</li><li>• ADD (1 バイト) 全てのバイトは unsigned 16 ビット値として足されます。結果の値の下位 8 ビットはチェックサムとして使用されます。</li><li>• AddInvASCII (2 バイト) 全てのバイトは unsigned 8 ビット値として足されます。結果の値の下位 8 ビットは反転されチェックサムとして使用され、16 進数の ASCII データ (2 バイト) として表示されます。</li></ul>

## コマンド

### 概論

前述したとおり、コマンドは保存と再利用が可能な予め設定されたトランザクションです。通常のトランザクションのように、コマンドはフレーム・オブジェクトを含みサブネットワーク上で通信される実際のシリアル・テレグラムを表示します。

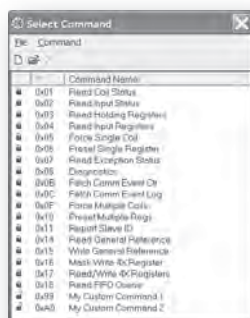
ノードへのコマンドの追加は結果的にコマンドで指定された方向に従い追加されたトランザクションになります。トランザクションのようなフレーム・オブジェクトは Parameter セクションの中のパラメータから値だけでなく、'SlaveAddress' パラメータのような他のデータも回収します (12-1 “ ノード・パラメータ ” を参照してください)。このような場合、Parameter セクションのパラメータはグレースアウトされ直接編集することができません。

マスター・モードでは、ABC コンフィグツールは最も一般的な Modbus RTU 機能を使用し予めロードされます。コマンドの追加はコマンド・エディタ (15-3 “ コマンド・エディタ ” を参照してください) を使用して簡単に行えます。ジェネリック・データ・モードでは、予めコマンドは設定されていませんがご要望のコマンドを実装することができます。

### コマンドの追加と管理

ノードへのコマンドの追加をするために Navigation セクション内のノード上で右クリックし、'Add Command' を選択します。

コマンドリストが表示されます：



ご要望のコマンドをリストから選び、'Command' メニューの中の 'Add Command' を選択します。指定したコマンドがノードに追加されます。

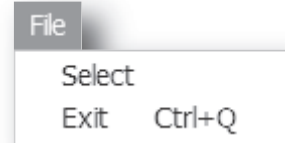
他のトランザクションのように、追加されたコマンドのフレーム・オブジェクトは Navigation/Parameter セクションまたはトランザクション・エディタを使用して編集することができます。あるフレーム・オブジェクトは編集することができないことに注意してください。

## プルダウン・メニュー

### File

このメニューには次のエントリがあります：

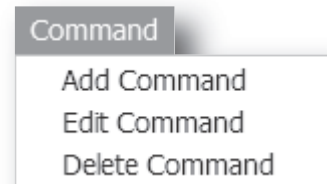
- **Select**  
選択されているコマンドをノードに追加します。
- **Exit**  
ノードにコマンドを追加せずにメニューを終了します。



### Command

このメニューはリスト内のコマンドを管理するために使用します：

- **Add Command**  
リストにカスタム・コマンドを追加しコマンド・エディタ内で新しいコマンドを開きます。  
15-3 “コマンド・エディタ” も参照してください。
- **Edit Command**  
コマンド・エディタを使用して現在選択されているコマンドを編集します。  
15-3 “コマンド・エディタ” もご参照ください。
- **Delete Command**  
リストから現在選択されているコマンドを削除します。削除できないコマンドもあることに注意してください。



## Toolbar Icons

ツールバーには最も一般的に使用される機能のアイコンがあります。

- **Add Command**  
(‘Command’ メニュー内の ‘Add Command’ を参照してください。)
- **Edit Command**  
(‘Command’ メニュー内の ‘Edit Command’ を参照してください。)
- **Delete Command**  
(‘Command’ メニュー内の ‘Delete Command’ を参照してください。)



# コマンド・エディタ

## 概論

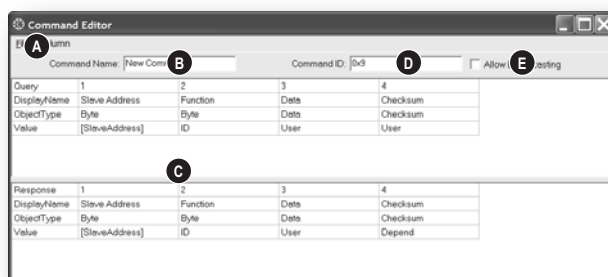
コマンド・エディタは新しいコマンドを設定し既存のコマンドを編集するために使用します。この機能により後に再利用することができるコマンドのライブラリを作成することができます。

フレーム・オブジェクトによっては削除または変更することができませんのでコマンド・エディタは多少プロトコル依存になります。

このセクションの例ではマスター・モードを使用します。手順は Modbus RTU プロトコルによる制約がないことを除いてはジェネラル・データ・モードの手順と似ています。

## 基本ナビゲーション

‘Edit Command’ または ‘Command’ メニューの ‘Add Command’ を選ぶことでコマンド・エディタを開きます。



### A: プルダウン・メニュー

15-4 “プルダウン・メニュー” を参照ください。

### B: コマンド名

テキスト形式のコマンド名です。

### C: コマンド・トランザクション

このセクションはコマンドに関連した実際のトランザクションが含まれます。このトランザクションはクエリ / レスポンスの一組または単体のトランザクションのいずれかの設定ができます。

### D: コマンド ID

コマンドを作成するときにご要望に応じて使用することができます。例えば機能コードを指定するために使用します。

### E: 他の設定

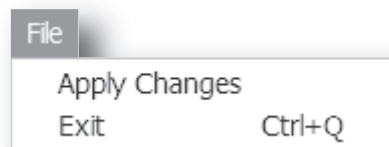
設定	説明
Allow Broadcasting	コマンドをブロードキャストできるかどうか指定します（マスター・モードのみ関連）
Produce	コマンドはプロデュースされるデータです（ジェネリック・データ・モードのみ）
Consume	コマンドはコンスームされるデータです（ジェネリック・データ・モードのみ）

## プルダウン・メニュー

### File

このメニューには次のエントリがあります：

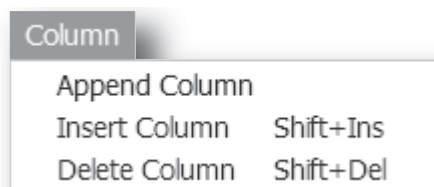
- **Apply Changes**  
変更を保存してメイン・ウィンドウに戻ります。
- **Exit**  
保存せずにメイン・ウィンドウに戻ります。



### Column

このメニュー内の機能はコマンドの構成を変更します。

- **Append Column**  
もう一つのカラムをコマンドに追加します。
- **Insert Column**  
選択した位置にカラムを挿入します。
- **Delete Command**  
選択した位置のカラムを削除します。





## コマンドの編集

前述したように、コマンド・エディタ内のトランザクション・セクションはコマンドに関連付けられた実際のトランザクションを表示します。各カラムはトランザクション内のフレーム・オブジェクトを表します。

各カラムには次のパラメータをもつ4つのカラムがあります：

- **Query/Response/Produce/Consume**  
右上のセルはトランザクションの方向を示します。
- **DisplayName**  
メイン・ウィンドウのトランザクション・エディタまたは **Parameter** セクション内で設定するときコマンドの異なる部分をより理解しやすい方法で表示するために各カラムに名前を付けることができます。
- **ObjectType**  
この列はカラムに使用するフレーム・オブジェクトの種類を指定します。
- **Value**  
この列はフレーム・オブジェクトが値 / 設定を収集する場所を指定します。

Value	説明
Depend	この設定はマスター・モードのレスポンスのみに関連します。 値は関連する 'Query' トランザクションの部分から回収されます。
Id	値は 'Command ID' 設定から回収されます (15-3 " 基本ナビゲーション " を参照してください)。
User	オブジェクトに関連する設定はユーザにより編集することができます。
[SlaveAddress]	値は 'SlaveAddress' パラメータから回収されます (12-1 " ノード・パラメータ " を参照してください)。
(other settings)	他の設定は現在サポートされていません。

## 例：マスター・モードでの Modbus-RTU コマンドの指定

次の例では Modbus-RTU コマンドはマスター・モードで作成されます。Modbus-RTU ではトランザクションは常に次の部分が使用されます。

- Slave Address (1 byte)
- Function Code (1 bytes)
- A data field
- CRC (CRC-16)

さらに各コマンドはクエリとレスポンスを含みます。

### • クエリの例

クエリ	1	2	3	4
DisplayName	Slave Address	Function	Data	Checksum
Object Type	Byte Object	Byte Object	Data Object	Checksum Object
Value	[SlaveAddress]	ID	User	User
	この Byte constant の値は 'SlaveAddress' パラメータを使用し設定されます (12-1 " ノード・パラメータ " を参照してください)。	この Byte constant の値は 'Command ID' フィールドを使用し設定されます。	ユーザにより決定されたこの Object に関連したデータの位置と大きさです。	チェックサムの種類などはユーザが選択することができます。デフォルトでは、Modbus-RTU 標準に一致させるためにこれが設定されます。

### • レスポンスの例

レスポンス	1	2	3	4
DisplayName	Slave Address	Function	Data	Checksum
Object Type	Byte Object	Byte Object	Data Object	Checksum Object
Value	[SlaveAddress]	ID	User	Depend
	この値は Parameter Window の 'SlaveAddress' パラメータにリンクします。	Byte Constant の値は 'Command ID' フィールドを使用し設定されます。	ユーザにより決定されたこの Object に関連するデータの大きさと位置です。	この Object はクエリ内の対応する Object から設定を回収します。

デフォルトで Modbus-RTU-specific フレーム・オブジェクトは既に決まった場所にありデータ・オブジェクトは機能コードと CRC の間に挿入されます。これらのオブジェクトは移動または削除することはできませんがご要望の機能コードと CRC の間にオブジェクトを追加することができます。

‘Command Name’ フィールド内にその名前を入力し新しいコマンドの名前を設定し、‘Command ID’ フィールド内にふさわしい機能コードを入力します。コマンドをブロードキャスト送信する場合には、‘Allow Broadcasting’ チェックボックスにチェックを入れます。

## サブネットワーク・モニタ

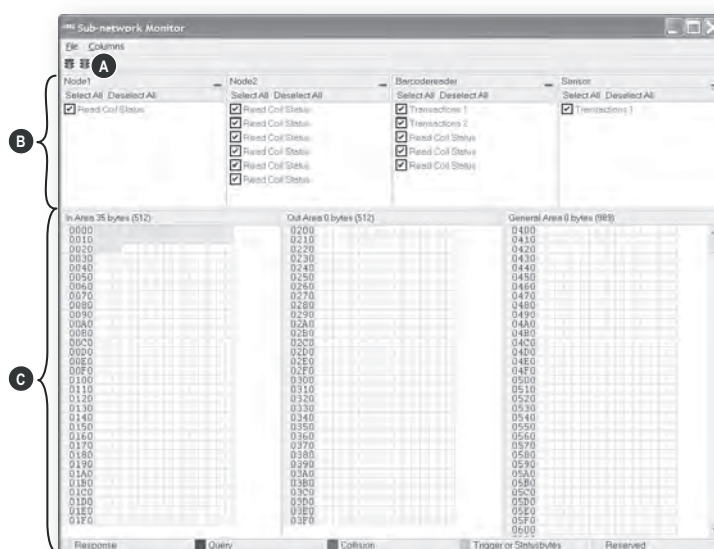
### 概論

サブネットワーク・モニタはサブネットワークのコンフィグレーションとトラブルシューティングを簡単にします。主な機能はサブネットワーク通信向けに割り当てられたデータの表示とコリジョンが発生した場合などの領域の割り当ての重複の検出を行います。

コンフィグレーションされた全てのノードとトランザクションはスクリーン (B) の中央にリストされます。一つのトランザクションの選択と選択解除をすることで、割り当てられたデータの組み合わせを見ることができます。

**注意：** サブネットワーク・モニタはゲートウェイ全体のパフォーマンスにネガティブな影響を与えます。モニタ機能は注意して使用してください。

### 操作



#### A: ネットワーク開始とネットワーク停止アイコン

これらのアイコンはサブネットワークの動作を制御します。サブネットワークの動作を止めるためには赤いライトをクリックします。サブネットワークを再び開始するには緑のライトをクリックします。



Start



Stop

#### B: ノード / トランザクション

トランザクションに関連したデータ・ブロックを見るためにはリストからトランザクションを選択します。対応するデータがモニタ・セクション (C) に表示されます。

#### C: モニタ・セクション

このセクションでは入出力と一般データ領域内のデータの割り当て方を表示します。

色	説明
白	割り当てられていません
黄	Response または Consume トランザクションによって割り当てられたデータ
青	Query または Produce トランザクションによって割り当てられたデータ
赤	コリジョン；一度以上の領域の割り当て
灰	リザーブド（メモリ消費の表示、必要な場合領域の割り当てが可能）
緑	Trigger byte、Transmit/Receive Counter、または Control/Status Register によって割り当てられたデータ

## ノード・モニタ

### 概論

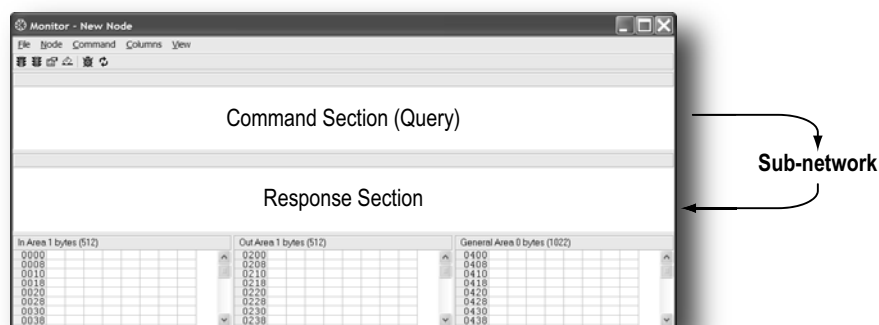
手動で個別のコマンドを送信できる設定を行い（使用できる場合）レスポンスをモニタすることでノード・モニタはサブネットワークの通信の設定のための情報を提供します。特定のノードによって使用されているメモリの概要も提供します。

**注意：** ノード・モニタはゲートウェイのパフォーマンス全体にネガティブな影響を与えます。必要なときだけ使用するようにしてください。

二つのプロトコル・モードでノード・モニタは幾分異なる動作をします。

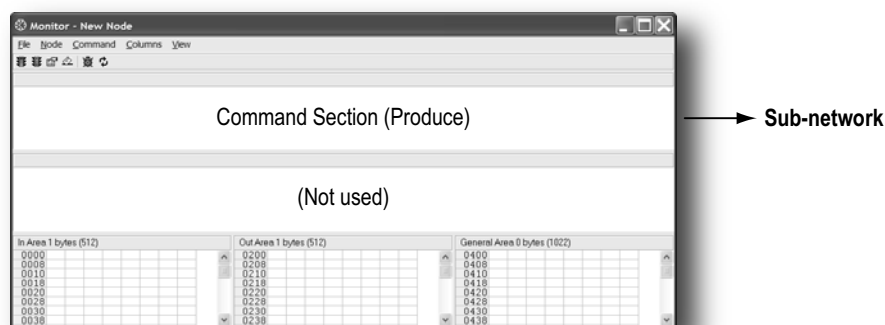
- マスター・モード

選択されたコマンド（Query Transaction）はサブネットワークへ送られます。クエリへのレスポンスは Response セクションで見ることができます。

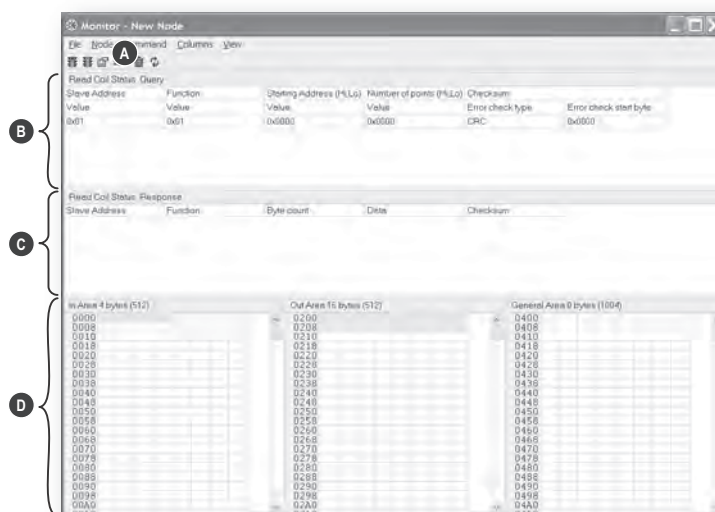


- ジェネリック・データ・モード

選択されたコマンド（Transaction Produce）はサブネットワークへ送られます。他のノードによって作成されたレスポンスなどを見ることはできません。



## ノード・モニタのナビゲーション



### A: プルダウン・メニューとツールバー・アイコン

17-3 “プルダウン・メニュー”と 17-4 “ツールバー・アイコン”を参照してください。

### B: コマンド・セクション

このセクションは選択されているレスポンスが表示されます。コマンド内の個々のフレーム・オブジェクトはトランザクション・エディタならびにコマンド・エディタで同様の方法で編集することができます。

### C: レスポンス・セクション (マスター・モードのみ)

このセクションは選択されたコマンドへのレスポンスが表示されます。

### D: モニタ・セクション

このセクションはノードに関連したデータを表示します。濃い灰色の領域は Status & Control Register のためにリザーブされ、薄い灰色の領域はノードにより使用されるデータを表します。

このセクションに表示されたデータはツールバー内の refresh アイコンに基づいてリフレッシュされます。詳細は 17-4 “ツールバー・アイコン”を参照してください。

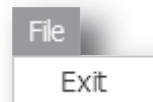
## プルダウン・メニュー

### File

このメニューには一つだけエントリがあります：

- **Exit**

このメニューは Node Monitor を閉じます。ノードが ‘Stop Node’ (下記参照してください) を使用し無効の場合、‘Start node’ を使用し有効にするまではデータ交換を行いません。



### Node

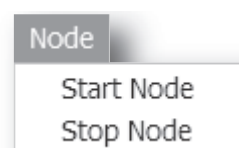
このメニューはノードのデータ交換を制御します。特定のノードに関連した問題を切り分けるために使用できます。

- **Start Node**

ノードに関連したトランザクションを有効にします。

- **Stop Node**

ノードに関連したトランザクションを無効にします。



### Command

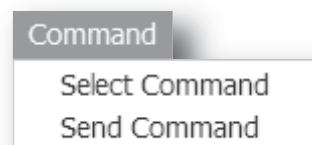
このメニューは手動でコマンドを指定し送信するために使用します。

- **Select Command**

サブネットワーク上に送信されるコマンドを選択します。

- **Send Command**

指定されたコマンドをサブネットワークへ送信します。



### Columns

このメニューは Monitor セクション内のコラム数を指定します。

- **Free**

コラム数はウィンドウの幅に依存します。

- **8 Multiple**

コラム数は 8 に固定されます。



### View

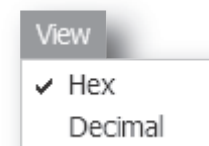
このメニューは Monitor セクション内でのデータの表現を指定します。

- **Hex**

16 進数フォーマットでデータの表示をします。

- **Decimal**

10 進数フォーマットでデータの表示をします。



## ツールバー・アイコン

ツールバーには最も一般的に使用する機能のためのアイコンがあります。

- ノードの開始とノードの停止

これらのアイコンは 'Node' メニューの中の機能に対応します。17-3 “Node” も参照してください。



Start



Stop

- コマンドの選択とコマンドの送信

これらのアイコンは 'Command' メニューの中の機能に対応します。17-3 “Command” も参照してください。



Select



Send

- リフレッシュの継続とリフレッシュの停止

有効に設定した場合、Monitor セクションに表示されたデータは周期的にリフレッシュされます。無効に設定した場合、つまりリフレッシュを停止した場合、データは 'Refresh' アイコンを使用して手動でリフレッシュされる必要があります（下を参照してください）。



Stop



Resume

- リフレッシュ

このアイコンをクリックすると、Monitor セクションに表示されたデータはリフレッシュされます。



Refresh

# データ・ロガー

## 概論

この機能を使用しサブネットワーク通信のログを検証のためバッファに保存します。この機能はサブネットワークの最下位のデバッグを行う場合に様々な情報を提供します。ログ機能はゲートウェイの一部で ABC コンフィグツールとは別です。

これはゲートウェイが ABC コンフィグツールが動作している PC と接続されていなくてもログをとることができることを意味します。

## 操作

### ログの開始と停止

- ログの開始 (Start logging)

‘Tools’ メニューから ‘Start Logging’ を選択します。ABC コンフィグツールは操作モードの選択を促します。下を参照してください。

- ログの停止 (Stop logging)

‘Tools’ メニューの中の ‘Stop Logging’ を選択します。Log window が開きます、下を参照してください。

### 操作のモード

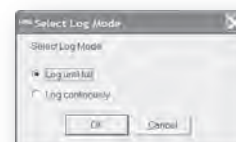
ご要望の操作のモードを選択し、‘OK’ をクリックしデータのログを開始します。

- Log until full

容量がフルになるまでログをとります。

- Log continuously

‘Stop Logging’ をクリックしログが停止されるまで、継続してログが取られます。ログ・バッファには最新データが残ります。

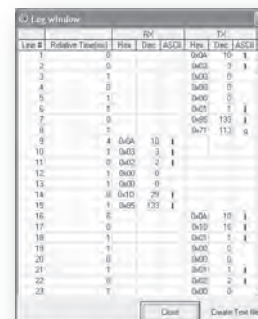


### Log Window

ログデータは双方向で 16 進数、10 進と ASCII フォーマットで表示されます。ログエントリ間の時間はカラムごとに表示されます。

‘Create Text file’ をクリックすることでデータを ASCII テキストフォーマットでセーブすることもできます。

ログを終了する場合は ‘Close’ をクリックします。





## コンフィグレーション

デフォルトでは、ログ・バッファは各方向に対して 512 バイト保存することができます。異なる大きさのバッファを指定するためには 'Tools' メニューの 'Options' を選択します。

様々な設定のウィンドウが表示されます。'ABC' タブを選択して 'Size of logbuffer' の下にバッファ・エントリを入力します（有効な設定値は 1...512 になります）。

新しい設定を有効にするため 'Apply' をクリックします。

'OK' をクリックし終了します。



# コンフィグレーション・ウィザード

## 概論

新たなサブネットワーク・コンフィグレーション実行時、ABC コンフィグツールでははじめからコンフィグレーションを開始するか事前定義されたテンプレート、つまりウィザードを使用するか、2つの選択肢があります。

ウィザードはユーザから提供される情報に基づいてサブネットワーク・コンフィグレーションを作成するため、ユーザが行うことは「ブランクを埋める」を埋めるだけです。サブネットワークがウィザード・プロファイルに合致する場合のみ機能することに注意してください。他の場合はすべて「ブランク・コンフィグレーション」オプションが使用されなければなりません。

## ウィザード・プロファイルの選択

ABC コンフィグツールが開始される毎に、あるいは「ファイル」File」メニューで「New」エントリを選択する毎に次のウィンドウが表示されます。（「Options」メニューで無効にされなければ、10-3 “Tools” を参照してください。）

現在次のウィザードが利用可能です。

- **ABCC ExtLink Wizard**

このウィザードは Anybus-CompactComModbus-RTUフィールドバス通信アダプタの使用向けです。

- **Wizard - Modbus RTU Master**

このオプションは ModbusRTU ベースのネットワークに最適です。

19-2 “ウィザード – Modbus RTU マスター” を参照してください。

- **Blank Configuration**

このオプションは空のコンフィグレーションを生成します。



希望するウィザードをハイライトし「OK」をクリックします。

## ウィザード – Modbus RTU マスター

このウィザードはサブネットワークについてのある情報に基づいたModbus-RTU ベースのネットワーク・コンフィグレーションを実行します。オンライン・ヘルプ・システムがコンフィグレーションの各ステップを詳細に説明します。

### • 重要な注意：

多くの機器が Modbus 標準に完全には対応していません。例えばこの標準を変更し実装する場合や、このウィザードで使用されているコマンド以外の特定の Modbus コマンドに限定し使用する場合です。これらすべての場合においてユーザは、サブネットワーク上で使用される機器のシリアル通信要件についての情報をドキュメントで確認していなければなりません。必要であればシリアル通信プロトコルについての詳細な情報を得るために機器メーカーに連絡する必要があります。

機器が要求する特定 Modbus コマンドをこのウィザードが処理しない場合、ABC コンフィグツールでこのコマンドをトランザクションとしてマニュアルで特定することができます。

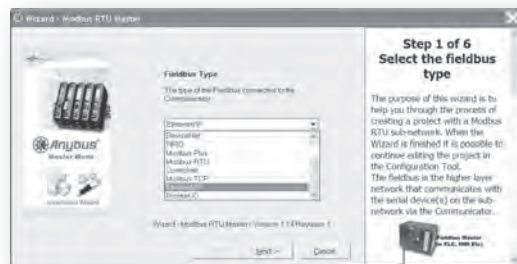
このウィザードの使用には次のステップが含まれています。

### ステップ 1: Communicator タイプ

‘EtherNet IP’ を選択します。

続けるために ‘Next’ をクリックします。

**ポイント:** ‘Previous’ をクリックすることでどんな設定もそのままいつでも前のメニューに戻ることができます。



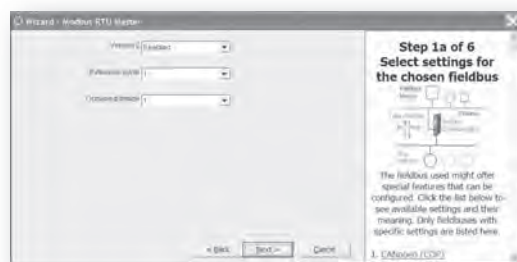
### ステップ 1a: I/O サイズ

このパラメータは入/出力データ領域のサイズを設定するために使用されます。

続けるために ‘Next’ をクリックします。

- 2-6 “EtherNet/IP”
- 11-1 “IO Sizes”

も参照してください。



### ステップ 2: 物理的な設定

サブネットワークの物理的なプロパティを選択します。

続けるために ‘Next’ をクリックします。



### ステップ 3-6

より詳細な情報についてはオンライン・ヘルプ・システムを参照してください。

## 制御 / ステータス・レジスタ

### 概論

制御 / ステータス・レジスタはデフォルトでは無効ですが、ABC コンフィグツール（11-3 “Status / Control Word” を参照してください）を使用して有効にすることができます。これらのレジスタはサブネットワークとフィールドバス制御システム間のステータス情報を交換するためのインターフェースを生成します。

これらのレジスタの主な目的は ...

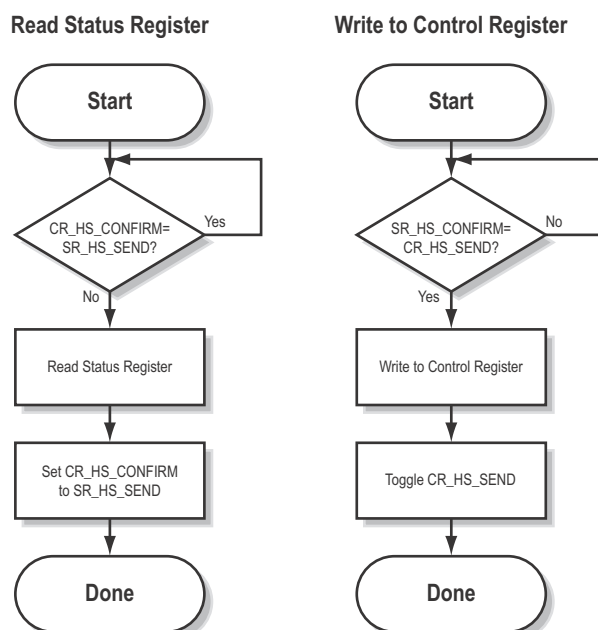
- フィールドバス制御システムへサブネットワークに関連する問題を報告する
- 双方向で有効なデータだけが交換されたことを確認する
- フィールドバス制御システムがサブネットワーク上の選択されたノードでデータ交換をスタート / 終了可能にする

有効であればこれらのレジスタは入出力領域の最初の 2 バイト（それぞれ 0x000-0x001 と 0x200-0x201）を占有します。そのためこれらの領域では他のデータのようにフィールドバスからアクセスすることが可能です。

**注意：** 内部ではこれらのレジスタは Motorola フォーマット（MSB）で保存されます。もし上位レベル・ネットワークが異なるバイト順を使用するとバイトの上位 / 下位はスワップされます。

### ハンドシェーク手順

両部分が正しい情報を確実に受信するため、これらのレジスタにアクセスするときには、下図のような 2 つのフローチャートになる特別ハンドシェーク手順に従う必要があります。



## データの一貫性

制御 / ステータス・レジスタの‘データ有効’ビットはスタートアップとフィールドバス・オフライン/オンライン・トランザクション中にデータの一貫性を保つために使用されます。

ABC コンフィグツールで‘ステータス / 制御ワード’パラメータが‘有効’に設定されていると、ゲートウェイはフィールドバス制御システムがサブネットワーク上でデータ交換を開始する前に制御レジスタで‘データ有効’ビットを設定するのを待ちます。

同じパラメータが‘無効’あるいは‘有効ただしロック解除なし’に設定されていると通信はフィールドバスがオンラインになると同時に開始します。

### ステートマシン

フィールドバス・ネットワークへの参加はステートマシンを使用して下記のように記述されます。

#### A: オフライン（データ交換なし）

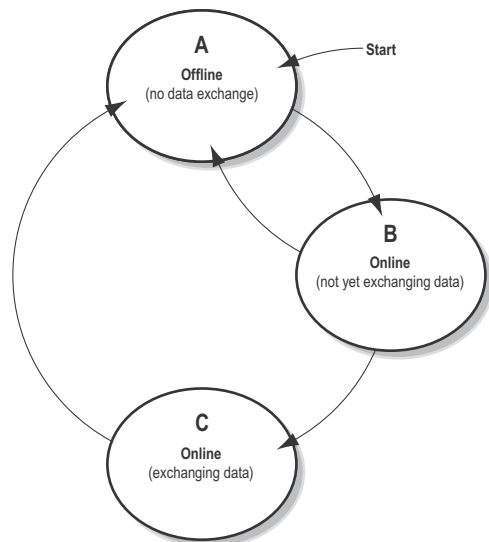
1. 制御レジスタでデータ有効’ビットをクリア
2. サブネットワーク・コンフィグレーションに従って出力領域に初期データを書き込み
3. フィールドバス・ネットワーク上でフィールドバス制御システムとゲートウェイがオンラインになるのを待ち、ステート B へ移行

#### B: オンライン（データ交換前）

4. ゲートウェイがステータス・レジスタで‘データ有効’ビットをクリアにするまで待つ
5. 制御レジスタで‘データ有効’ビットを設定
6. ゲートウェイがステータス・レジスタで‘データ有効’ビットを設定すればステート C へ移行
7. フィールドバス上でゲートウェイがオフラインになるとステート A へ移行

#### C: オンライン（データ交換中）

有効データを双方向で交換します。  
ゲートウェイがフィールドバス上でオフラインになればステート A へ移行



**注意：**ゲートウェイはステータス・レジスタで‘データ有効’ビットを自然にクリアすることはできません。

### 待ち時間

ステータス・レジスタの‘データ有効’ビットは遅延を引き起こすかもしれません。この待ち時間はノードがないか割り付けられている長いタイムアウト値でのノードへの接続が原因により発生します。

従ってフィールドバス制御システムはサブネットワーク機器で通信する前にこのビットが設定されるのを待ってはいけません。すべてのデータが更新された時、フィールドバス制御システムに知らせるビットとみなされます。

## ステータス・レジスタ構成（ゲートウェイから制御システムへ）

### 概論

ステータス・レジスタは（有効な場合）0x000-0x001 に位置し、次のようにビットフィールドを設定します。

ビット	名前	説明
15	送信 (SR_HS_SEND)	これらのビットはフィールドバス制御システムのハンドシェークを制御します。
14	確認 (SR_HS_CONFIRM)	
13	データ有効 (マスター・モードのみ)	このビットはすべてのトランザクションが一度でも実行できた時に設定されます。 一度設定されると変更されません。 1: データ有効 0: データ無効 <b>注意:</b> このビットはジェネリック・データ・モードでは使用されません。
12... 8	ステータス・コード	このフィールドはゲートウェイからの最新ステータス・レポートを保持しています。  - 20-3 " マスター・モードでのステータス・コード " - 20-4 " ジェネリック・データ・モードでのステータス・コード " も参照してください。
7... 0	データ	

**注意:** 内部ではこれは Motorola フォーマット・ワード (MSB ファースト) として扱われます。上位レベル・ネットワークが異なるバイト順を使用するとバイトの上位 / 下位はスワップされて表示されます。

### マスター・モードでのステータス・コード

(このテーブルはマスター・モードのみ有効です。)

コード	状態	タイプ	データ	説明
0x00	再伝送 カウンタ更新	警告	カウンタ	サブネットワーク上の再伝送数が増えました。もしこの問題が続くようであればシングルあるいはマルチノードが見当たらないのかも知れません。
0x01	シングルノードが見当たらない	エラー	スレーブ・アドレス	シングルノードが見当たりません。
0x02	マルチノードが見当たらない	エラー	ノード数	マルチノードが見当たりません。
0x03	バッファ・オーバーラン	警告	スレーブ・アドレス	ノードが予想以上のデータを返信します。
0x04	その他エラー	エラー	スレーブ・アドレス	定義のないエラー
0x1F	エラー無し	警告	-	エラー無し

**注意:** タイプ 'エラー' は原因が解決したときに 'エラー無し' 状態になります。タイプ '警告' は情報として見なされ後に 'エラー無し' 状態になる必要がないかも知れません。

## ジェネリック・データ・モードでのステータス・コード

(このテーブルはジェネリック・データ・モードでのみ有効です。)

コード	状態	タイプ	データ	説明
0x00	無効トランザクション・カウンタ更新	エラー	カウンタ	無効トランザクション数(サブネットワーク・コンフィギュレーションで定義されたどのコンスーム・トランザクションにも一致しないトランザクションの受信)が増加しています。
0x01	フレーム・エラー	警告	-	終端キャラクタは有効ですが、受信前にメッセージ・デリミタ・タイムアウトが発生しました。
0x02	オフライン・タイムアウト・カウンタ更新	エラー	カウンタ	タイムアウト・コンスーム・トランザクション数が増加しています。  - 13-4 "コンスーム・トランザクション (Consume-Transactions)"(オフライン・タイムアウト時間)も参照してください。
0x03	バッファ・オーバーラン	警告	-	ノードが予想より多いデータを返信、あるいはゲートウェイが新しいメッセージを受信する前に処理できません。
0x04	その他エラー	エラー	-	定義されていないエラー
0x1F	エラー無し	警告	-	エラーなし

**注意:** タイプ 'エラー' は原因がこれ以上検索されない時に 'エラー無し' になることがあります。'警告' は情報として見なされ後に 'エラー無し' にならないかもしれません。

## 制御レジスタ構成（制御システムからゲートウェイへ）

### 概論

制御レジスタは（有効であれば）0x200-0x201 に位置し、次のようにビットフィールドを設定します。

ビット	名前	説明
15	確認 (CR_HS_CONFIRM)	これらのビットはフィールドバス制御システムのハンドシェークを制御します。 - 20-1 " ハンドシェーク手順 " - 20-3 " ステータス・レジスタ構成（ゲートウェイから制御システムへ） " も参照してください。
14	送信 (CR_HS_SEND)	
13	データ有効	このビットはデータの一貫性を制御します。（20-2 " データの一貫性 " を参照してください。） 1: 出力領域有効；サブネットワーク上のデータ交換可能 0: 出力領域無効；サブネットワーク上でのデータ交換不可 <b>注意：</b> このビットは制御 / ステータス・レジスタが '有効' な場合のみ該当します。
12	実行コマンド	設定されると特定コマンドがゲートウェイにより実行されます。（下記参照してください。）
11... 8	制御コード	このフィールドはゲートウェイに実行可能なコマンドを保持しています。（下記参照してください。）
7... 0	データ	- 20-5 " マスター・モードでの制御コード " - 20-5 " ジェネリック・データ・モードでの制御コード " も参照してください。

**注意：**内部ではこれはモトローラ・フォーマット・ワード（MSB ファースト）として扱われます。上位レベル・ネットワークが異なるバイト順を使用するとバイトの上位 / 下位はスワップされて表示されます。

### マスター・モードでの制御コード

（このテーブルはマスター・モードでのみ有効です。）

コード	指示	データ	説明
0x00	無効ノード	実ノード・アドレス	特定ノードを無効にします。
0x01	有効ノード	実ノード・アドレス	先に無効にされたノードを有効にします。
0x02	有効ノード	実イネーブル・ノード数	特定されたノード数を有効にし、コンフィギュレーションでの最初のノードから開始します。残りのノードは無効になります。

### ジェネリック・データ・モードでの制御コード

（このモードでは現在制御コードをサポートしていません。）



# CIP オブジェクトの実装

## 概論

次の CIP オブジェクトがこの製品に実装されています。

### 必須オブジェクト

オブジェクト	ページ
識別オブジェクト、クラス 01h	21-2
メッセージ・ルータ、クラス 02h	21-3
アセンブリ・オブジェクト、クラス 04h	21-4
ポート・オブジェクト、クラス F4h	21-8
TCP/IP インターフェース・オブジェクト、クラス F5h	21-9
Ethernet リンク・オブジェクト、クラス 6Fh	21-10

### ベンダ特有オブジェクト

オブジェクト	ページ
診断オブジェクト、クラス AAh	21-5
パラメータ・データ入力マッピング・オブジェクト、クラス B0h	21-6
パラメータ・データ出力マッピング・オブジェクト、クラス B1h	21-7

## 識別オブジェクト、クラス 01h

### 概論

#### オブジェクト説明

-

#### サポートされるサービス

クラス・サービス :       Get Attribute All  
                                   Get Attribute Single

インスタンス・サービス :   Get Attribute All  
                                   Get Attribute Single  
                                   Reset

### クラス属性

#	アクセス	名前	タイプ	値	説明
1	Get	Revision	UINT	0001h	Revision 1

### インスタンス属性

#	アクセス	名前	タイプ	値	説明
1	Get	Vendor ID	UINT	デフォルト : 005Ah	HMS Industrial Networks AB
2	Get	Device Type	UINT	デフォルト : 000Ch	通信アダプタ
3	Get	Product Code	UINT	デフォルト : 0002h	DeviceNet 向け Anybus Communicator
4	Get	Revision	Struct of:		-
			USINT		メジャー・フィールドバス版
			USINT		マイナー・フィールドバス版
5	Get	Status	WORD	-	機器ステータス、下記テーブル参照ください
6	Get	Serial Number	UDINT	モジュール・シリアル番号	モジュールのシリアル番号
7	Get	Product Name	SHORT_STRING	'Anybus-C EtherNet/IP'	製品名

機器ステータス

bit(s)	名前
0	モジュール所有
1	(リザーブド)
2	コンフィグレーション済み
3	(リザーブド)
4... 7	拡張された機器ステータス : <div> <div>値:</div> <div>意味:</div> <div>0000b 不明</div> <div>0010b I/O 接続の失敗</div> <div>0011b I/O 未接続</div> <div>0100b 不揮発性コンフィグレーション失敗</div> <div>0110b 接続実行モード</div> <div>0111b 接続休止モード</div> <div>(その他) (リザーブド)</div> </div>
8	マイナーな復旧できるフォルトのための設定
9	マイナーな復旧できないフォルトのための設定
10	メジャーな復旧できるフォルトのための設定
11	メジャーな復旧できないフォルトのための設定
12... 15	(リザーブド)

# メッセージ・ルータ、クラス 02h

## 概論

### オブジェクト説明

-

### サポートされるサービス

クラス・サービス : -  
 インスタンス・サービス : -

## クラス属性

-

## インスタンス属性

-

## アセンブリ・オブジェクト、クラス 04h

### 概論

#### オブジェクト説明

このオブジェクトで ABC の入力 / 出力データ領域の I/O データへアクセスできます。

- 2-6 “EtherNet/IP”
- 11-1 “フィールドバス設定”

も参照してください。

#### サポートされるサービス

クラス・サービス : Get Attribute Single

インスタンス・サービス : Get Attribute Single  
Set Attribute Single

### クラス属性

#	アクセス	名前	タイプ	値	説明
1	Get	Revision	UINT	0002h	Revision 2
2	Get	Max Instance	UINT	-	The highest initiated instance no.

### インスタンス属性 – インスタンス / 接続ポイント 64h

このインスタンスは ABC メモリ内の I/O データ（入力）のアクセスに使用されます。

**注意:** I/O データ入力サイズが 0 に設定されている場合このインスタンスは初期化されません。

#	アクセス	名前	タイプ	値	説明
3	Get	Data	Array of BYTE	-	ABC により作成されたデータ

### インスタンス属性 - Instance/Connection Point 96h

**注意:** I/O 出力データサイズが 0 に設定されている場合このインスタンスは初期化されません。

#	アクセス	名前	タイプ	値	説明
3	Set	Data	Array of BYTE	-	ABC により消費されるデータ <sup>a</sup>

- a. Rockwell Automation 社製の PLC は最初の 4 バイトはステータス情報として定義され消費されます。この同社は Rockwell Automation 社製の機器により指定され EtherNet/IP 仕様で定義されるものではありません。多くの PLC はこのように実装されているため、ABC はこの動作を採用し消費されるデータから最初の 4 バイトを取り除いています。

## 診断オブジェクト、クラス AAh

### 概論

#### オブジェクト説明

このベンダ特有のオブジェクトはモジュールからの診断情報を提供します。

#### サポートされるサービス

クラス・サービス : Get Attribute All

インスタンス・サービス : Get Attribute Single

### クラス属性

#	アクセス	名前	タイプ	値	説明
1	Get	Revision	UINT	0001h	Revision 1

### インスタンス属性 , Instance 01h

#	アクセス	名前	タイプ	説明
01h	Get	Module serial number	UDINT	シリアル番号
02h	Get	Vendor ID	UINT	製造メーカー ID 番号
03h	Get	Fieldbus Type	UINT	フィールドバスの種類
04h	Get	Module Software version	UINT	モジュールのソフトウェアのバージョン
0Ah	Get	Module Type	UINT	モジュールの種類
0Fh	Get	IN cyclic I/O length	UINT	I/O 入力領域のサイズ (バイト)
11h	Get	IN total length	UINT	サポートされる入力の合計バイト数
12h	Get	OUT cyclic I/O length	UINT	I/O 出力領域のサイズ (バイト)
14h	Get	OUT total length	UINT	サポートされる出力の合計バイト数

# パラメータ・データ入力マッピング・オブジェクト、 クラス B0h

## 概論

### オブジェクト説明

このオブジェクトは入力データにアサイクリックにアクセスするために使用され、パラメータ・データ・メールボックスの初期化に基づいて動的に設定されます。(A-1 “パラメータ・データの初期化 (エクスプリシット・データ)” を参照してください。)

- 2-6 “EtherNet/IP”
- 11-1 “フィールドバス設定”
- 21-7 “パラメータ・データ出力マッピング・オブジェクト、クラス B1h”
- A-1 “パラメータ・データの初期化 (エクスプリシット・データ)”

も参照してください。

### サポートされるサービス

クラス・サービス : Get Attribute All

インスタンス・サービス : Get Attribute Single

## クラス属性

#	アクセス	名前	タイプ	値	説明
1	Get	Revision	UINT	0001h	Revision 1

## インスタンス属性、インスタンス 01h

各属性は入力データのブロックに対応します。各ブロックのサイズと割り当ては ABC コンフィグルールで設定されなければなりません。

詳細については A-1 “パラメータ・データの初期化 (エクスプリシット・データ)” を参照してください。

#	アクセス	名前	タイプ	説明
01h	Get	Data	Array of USINT	入力データの場合マッピングされたブロック
02h	Get	Data	Array of USINT	入力データの場合マッピングされたブロック
02h	Get	Data	Array of USINT	入力データの場合マッピングされたブロック
02h	Get	Data	Array of USINT	入力データの場合マッピングされたブロック
02h	Get	Data	Array of USINT	入力データの場合マッピングされたブロック
02h	Get	Data	Array of USINT	入力データの場合マッピングされたブロック
...	...	...	...	...
32h	Get	Data	Array of USINT	入力データの場合マッピングされたブロック

# パラメータ・データ出力マッピング・オブジェクト、クラス B1h

## 概論

### オブジェクト説明

このオブジェクトは出力データにアサイクリックにアクセスするために使用され、パラメータ・データ・メールボックスの初期化に基づいて動的に設定されます。(A-1 “パラメータ・データの初期化 (エクスプリシット・データ) ”) を参照してください。

- 2-6 “EtherNet/IP”
- 11-1 “フィールドバス設定”
- 21-6 “パラメータ・データ入力マッピング・オブジェクト、クラス B0h”
- A-1 “パラメータ・データの初期化 (エクスプリシット・データ) ”

も参照してください。

### サポートされるサービス

クラス・サービス : Get Attribute All

インスタンス・サービス : Get Attribute Single  
Set Attribute Single

## クラス属性

#	アクセス	名前	タイプ	値	説明
1	Get	Revision	UINT	0001h	Revision 1

## インスタンス属性、インスタンス 01h

各属性は出力データのブロックに対応します。各ブロックのサイズと割り当ては ABC コンフィグツールで設定されなければなりません。

詳細については、A-1 “パラメータ・データの初期化 (エクスプリシット・データ) ” を参照してください。

#	アクセス	名前	タイプ	説明
01h	Get/Set	Data	Array of USINT	出力データのマッピングされたブロック
02h	Get/Set	Data	Array of USINT	出力データのマッピングされたブロック
01h	Get/Set	Data	Array of USINT	出力データのマッピングされたブロック
02h	Get/Set	Data	Array of USINT	出力データのマッピングされたブロック
01h	Get/Set	Data	Array of USINT	出力データのマッピングされたブロック
02h	Get/Set	Data	Array of USINT	出力データのマッピングされたブロック
...	...	...	...	...
32h	Get/Set	Data	Array of USINT	出力データのマッピングされたブロック

## ポート・オブジェクト、クラス F4h

### 概論

#### オブジェクト説明

-

#### サポートされるサービス

クラス・サービス :       Get Attribute All  
                               Get Attribute Single

インスタンス・サービス :   Get Attribute All  
                               Get Attribute Single

### クラス属性

#	アクセス	名前	タイプ	値	説明
1	Get	Revision	UINT	0001h	インスタンスの最高番号 2
2	Get	Max Instance	UINT	0002h	インスタンス 1 が実装されました
3	Get	No. of instances	UINT	0001h	ポートを定義するポート・オブジェクトのインスタンスの返信
8	Get	Entry Port	UINT	0002h	各インスタンスからの属性 1 と 2 を含むアレイ・ストラクチャ。インスタンス 1 はオフセット 4 バイト、インスタンス 2 はオフセット 8 バイトになります。オフセット 0 の 4 バイトは 0 になります。(デフォルト)
9	Get	All Ports	Array of STRUCT {UINT; UINT;}	0000h 0000h 0000h 0000h 0004h 0002h	インスタンスの最高番号 2

### インスタンス属性、インスタンス 02h

#	アクセス	名前	タイプ	値	説明
1	Get	Port Type	UINT	0004h	TCP/IP
2	Get	Port Number	UINT	0002h	Port 2
3	Get	Port Object	Struct of:		
		Path Size	UINT	0002h	-
		Path	Padded EPATH	20 F5 24 01h	TCP クラス、インスタンス 1
4	Get	Port Name	SHORT_STIRNG	'TCP/IP'	ポート名
8	Get	Node Address	Padded EPATH	-	-



# TCP/IP インターフェース・オブジェクト、クラス F5h

## 概論

### オブジェクト説明

このオブジェクトは TCP/IP に関連する設定をグループ化します。

- 4-1 “基本的なネットワーク・コンフィグレーション”
- 11-1 “フィールドバス設定”

も参照してください。

### サポートされるサービス

クラス・サービス :           Get Attribute All  
                                  Get Attribute Single

インスタンス・サービス :   Get Attribute All  
                                  Get Attribute Single  
                                  Set Attribute Single

## クラス属性

#	アクセス	名前	タイプ	値	説明
1	Get	Revision	UINT	0001h	Revision 1
2	Get	Max Instance	UINT	0001h	インスタンスの最高番号 1
3	Get	No. of instances	UINT	0001h	インスタンス 1 が実装されました

## インスタンス属性

#	アクセス	名前	タイプ	値	説明
1	Get	Status	DWORD	00000001h	属性 #5 に無効な情報が含まれています。
2	Get	Configuration Capability	DWORD	00000014h	- 属性 #5 は設定可能 -DHCP を介したネットワーク・コンフィグレーション機能
3	Get/Set	Configuration Control	DWORD	-	値:   説明: 0   不揮発性メモリからのコンフィグレーション 2   DHCP からのコンフィグレーション
4	Get	Port Object	Struct of:		
		Path Size	UINT	0002h	2 words
		Path	Padded EPATH	20 F6 24 01h	Path to Ethernet Class, Instance 1
5	Get/Set	Interface Configuration	Struct of:		
		IP Address	UDINT	-	IP アドレス
		Subnet Mask	UDINT	-	サブネット・マスク
		Gateway Address	UDINT	-	ゲートウェイ・アドレス
		Name Server 1	UDINT	-	第一 DNS
		Name Server 2	UDINT	-	第二 DNS
		Domain Name	STRING	-	デフォルトでのドメイン名
6	Get/Set	Host Name	STRING	-	ホスト名

## Ethernet リンク・オブジェクト、クラス 6Fh

### 概論

#### オブジェクト説明

このオブジェクトは Ethernet インターフェースのための診断情報をグループ化します。

- 4-1 “基本的なネットワーク・コンフィグレーション”

も参照してください。

#### サポートされるサービス

クラス・サービス :           Get Attribute All  
                                  Get Attribute Single

インスタンス・サービス :   Get Attribute All  
                                  Get Attribute Single

### クラス属性

#	アクセス	名前	タイプ	値	説明
1	Get	Revision	UINT	0001h	Revision 1
2	Get	Max Instance	UINT	0001h	インスタンスの最高番号 1
3	Get	No. of instances	UINT	0001h	インスタンス 1 が実装されました

## インスタンス属性

#	アクセス	名前	タイプ	値	説明
1	Get	Interface Speed	UDINT	10 または 100	実際の ethernet インターフェースのスピード
2	Get	Interface Flags	DWORD	-	-
3	Get	Physical Address	Array of 6 USINTS	(MAC ID)	物理的なネットワーク・アドレス
4	Get	Interface Counters	Struct:		
		In Octets	UDINT	-	インターフェース上で受信した Octet
		In Ucast Packets	UDINT	-	インターフェース上で受信した Unicast パケット
		In NUcast Packets	UDINT	-	インターフェース上で受信した Non-unicast パケット
		In Discards	UDINT	-	未確認プロトコルを使用した Inbound パケット
		In Errors	UDINT	-	エラーを含む Inbound パケット (discards は含まない)
		In Unknown Protos	UDINT	-	未確認プロトコルを使用した Inbound パケット
		Out Octets	UDINT	-	インターフェース上で送信された Octet
		Out Ucast Packets	UDINT	-	インターフェース上で送信された Unicast パケット
		Out NUcast Packets	UDINT	-	インターフェース上で送信された Non-Unicast パケット
		Out Discards	UDINT	-	未確認プロトコルを使用した Inbound パケット
		Out Errors	UDINT	-	エラーを含む Outbound パケット (discards は含まない)
5	Get	Media Counters	Struct:		
		Alignment Errors	UDINT	-	Octets 長が正確でないフレームの受信
		FCS Errors	UDINT	-	受信した FCS チェックを伝えないフレーム
		Single Collisions	UDINT	-	一つの衝突で伝送されたフレーム
		Multiple Collisions	USINT	-	一つ以上の衝突で伝送されたフレーム
		SQE Test Errors	UDINT	0	-
		Deferred Transmissions	UDINT	-	媒体がビジー状態のため遅延傾向にある最初のフレームの伝送
		Late Collisions	UDINT	-	一つのパケットの伝送で 512 ビット回以降検出された衝突の回数
		Excessive Collisions	UDINT	-	例外的衝突のために伝送されなかったフレーム
		MAC Transmit Errors	UDINT	-	内部 MAC サブレイヤがエラーを受信したため伝送がうまくいかなかったフレーム
		Carrier Sense Errors	UDINT	-	フレームが伝送されるとき通信状態が失われるあるいは二度と使用できなくなる回数
		Frame Too Long	UDINT	-	許可された最大フレーム・サイズを超えたフレームの受信
		MAC Receive Errors	UDINT	-	内部サブレイヤがエラーを受信したためインターフェース上での受信がうまくいかなかったフレーム

# 先進フィールドバス・コンフィグレーション

## 概論

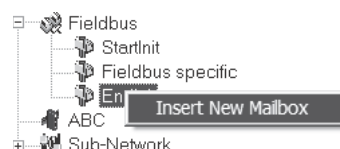
ゲートウェイのフィールドバス・インターフェースは組み込み Anybus-S 通信インターフェースで構成されています。通常、Anybus-S コンフィグレーション設定はゲートウェイにより自動的にセットアップされます。一方、先進ユーザは Anybus-S カードを特定機能向けにコンフィグレーションすることができます。この章では Anybus-S とそのアプリケーション・インターフェースに慣れているユーザを対象にしています。Anybus-S プラットフォームについての詳細は Anybus-S Parallel Design Guide を参照してください。

標準的な初期化パラメータはサブネットワーク・コンフィグレーションによって決定されます。サブネットワーク通信のための入力と出力データの合計についての情報は ABC コンフィグツールで使用されます。この情報によって ABC コンフィグツールは、ゲートウェイに組み込まれている Anybus-S インターフェースのデュアルポート RAM の入力と出力データ領域の大きさを設定するコンフィグレーション・メッセージを作成することができます。初期化をカスタマイズするためにフィールドバス特定メールボックス・メッセージを追加することも可能です。これはメールボックス・エディタで行われます。下記参照してください。

(メールボックス・メッセージは Anybus-S インターフェースを使用したローレベル通信向けに使用される HMS 特定コマンド・ストラクチャです。Anybus-S Parallel Design Guide と使用するフィールドバスのフィールドバス・アペンディックスを参照してください。)

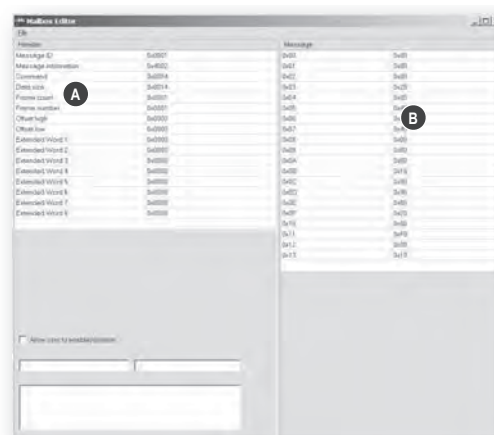
## メールボックス・エディタ

コンフィグレーションにメールボックス・メッセージを追加するためには、'EndInit' 上を右クリックして 'Insert New Mailbox' を選択してください。



メールボックス・メッセージはヘッダ部分とデータ部分で構成され、ヘッダは 16 ワード (32 バイト)、データは 128 ワード (256 バイト) 以下で構成されます。すべてのフィールドは 0x4002 で固定されているメッセージ情報フィールドを除いては編集可能で、ここではフィールドバス特定メールボックス・メッセージのみが入力することができます。

メールボックス・メッセージはふたつのコラムとして表示されます；ひとつはヘッダ情報を含む (A)、もうひとつはメッセージ・データを含む (B) です。



メッセージ・データを追加するためには、単にヘッダ・コラム (A) でデータサイズ・パラメータを変更します。メッセージ・データ・コラム (B) で対応するバイト数が表示されます。

特定メールボックス・メッセージの詳細については使用しているフィールドバスの "Anybus-S Fieldbus Appendix" を参照してください。Anybus-S プラットフォームの概論については "Anybus-S Design Guide" を参照してください。

# パラメータ・データの初期化（エクスプリシット・データ）

## 概論

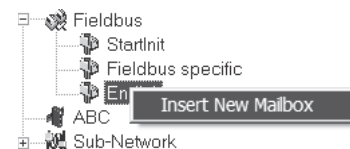
パラメータ・データとして宣言された入力 / 出力データの一部は適切な初期化がされない  
とネットワークからアクセスできません。

この手順の目的は、パラメータ・データ入力マッピング・オブジェクトとパラメータ・  
データ出力マッピング・オブジェクトのインスタンス属性に関連した入力 / 出力データ領  
域のデータ・ブロックを指定することです。

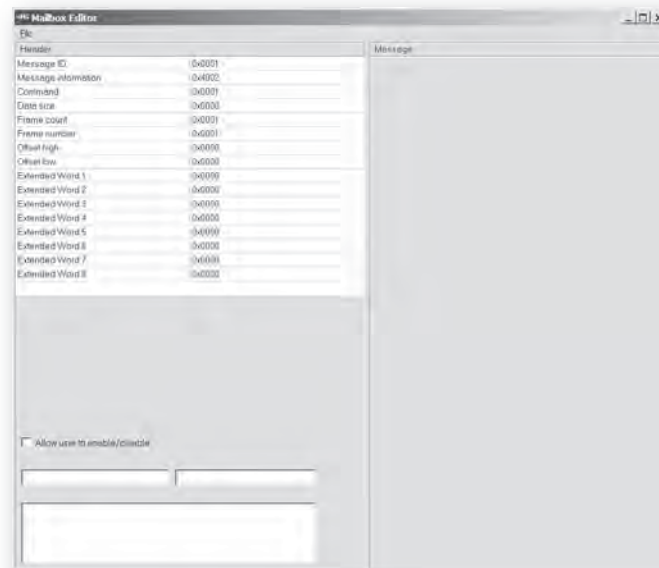
これを実現するため、ABC コンフィグツールのメールボックス・エディタ内で二つのメー  
ルボックス・メッセージを設定する必要があります。メールボックス・エディタについて  
の詳細については 22-1 “先進フィールドバス・コンフィグレーション” を参照してくださ  
い。

## メールボックス・メッセージの追加

メールボックス・メッセージをコンフィグレーション  
に追加するため、'EndInit' 上で右クリックし、'Insert New  
Mailbox' を選択してください。



次のウィンドウが表示されます。



このウィンドウ、つまり  
メールボックス・エディタ  
を使用した例は本章で後  
述されます。

22-1 “メールボックス・エ  
ディタ” も参照してくださ  
い。

## Ethernet/IP への入力パラメータ・データのマッピング

### 例

次の例では、データの 160 バイトはパラメータ・データ入力マッピング・オブジェクトにマッピングされます。データは 5 分割されたデータ・ブロックで、それぞれ特殊なインスタンス属性と関連付けられています。

これを実現するために次のステップを実行します。

1. コンフィグレーションに新しいメールボックス・メッセージを追加します。(A-1 “メールボックス・メッセージの追加” を参照してください。)
2. メールボックス・ヘッダ内の ‘Command’- 値を 0084h へ変更します。
3. メールボックス・ヘッダ（左カラム）内の ‘Data Size’ 値を調整します。この例では、各マッピングされた属性はメールボックス・データの 4 バイトを占有するためにサイズを 20 (0014h) に設定します。
4. メールボックス・データ・セクションの属性のためのマッピング割り付けを指定します。上述した通り、各マッピング入力には 4 バイトが必要です。データ・ブロックのオフセット<sup>1</sup>を指定する 2 バイト、データ・ブロックの長さを指定する 2 バイトに続きます。これらの値はビッグ・エンディアン（モトローラ）フォーマットで入力されなければなりません。

この例では次のメールボックス・データを提供します。

メールボックス・データ		属性番号	コメント
割り付け	データ		
0x00	0x00	1	Offset = 0000h
0x01	0x00		サイズ = 32 バイト
0x02	0x00		
0x03	0x20		
0x04	0x00	2	Offset = 0040h
0x05	0x40		サイズ = 64 バイト
0x06	0x00		
0x07	0x40		
0x08	0x00	3	Offset = 0080h
0x09	0x80		サイズ = 16 バイト
0x0A	0x00		
0x0B	0x10		
0x0C	0x00	4	Offset = 0090h
0x0D	0x90		サイズ = 32 バイト
0x0E	0x00		
0x0F	0x20		
0x10	0x00	5	Offset = 00F0h
0x11	0xF0		サイズ = 16 バイト
0x12	0x00		
0x13	0x10		

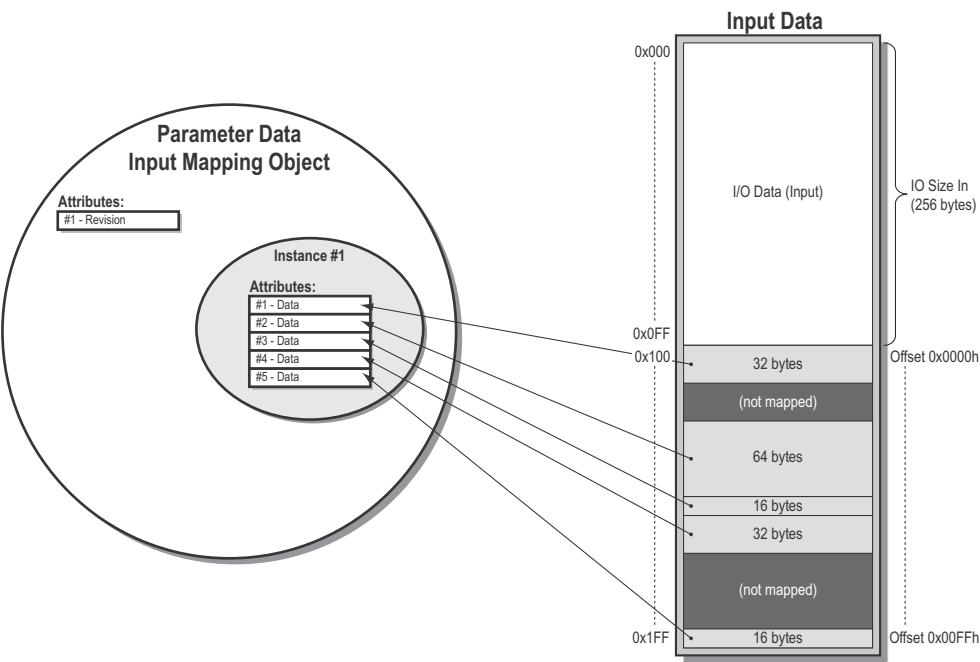
上記テーブルのように属性はマッピングされた順番で番号がつけられます。物理的にメールボックス・データのマッピング順を変更すると属性の番号も変更することができます。

5. 新しいメールボックスを保存するために ‘File’ メニューの ‘Apply changes’ を選択します。

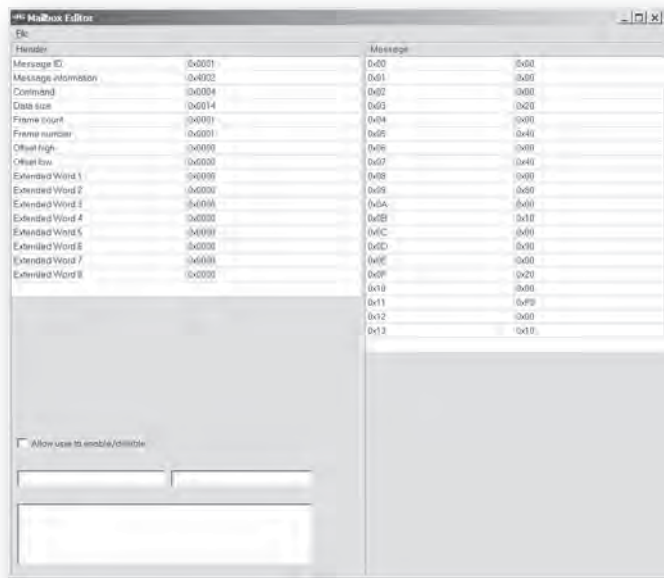
---

1. オフセットは ABC の物理メモリ割り付けではなく、パラメータ・データの開始から指定されます。

属性マッピングの結果



メールボックス・エディタ・スクリーンショット



## Ethernet/IP への出力パラメータ・データのマッピング

### 例

出力データのマッピングは入力データのマッピングに似ています。次の例では合計 144 バイトのデータがパラメータ・データ出力マッピング・オブジェクトにマッピングされています。データは 4 分割されたデータ・ブロックでそれぞれ特殊なインスタンス属性に関連づけられています。

これを実現するために次の手順を実行します：

1. 新しいメールボックス・メッセージをコンフィグレーションに追加します。(A-1 “メールボックス・メッセージの追加” を参照してください。)
2. メールボックス・ヘッダ内の ‘Command’ 値を 0085h へ変更します。
3. メールボックス・ヘッダ（左カラム）内の ‘Data Size’ 値を調整します。この例では、各マッピングされた属性はメールボックス・データの 4 バイトを占有するためにサイズを 20 (0014h) に設定します。
4. メールボックス・データ・セクションの属性のためのマッピング割り付けを指定します。上述した通り、各マッピング入力には 4 バイトが必要です。データ・ブロックのオフセット<sup>1</sup>を指定する 2 バイト、データ・ブロックの長さを指定する 2 バイトに続きます。これらの値はビッグ・エンディアン（モトローラ）フォーマットで入力されなければなりません。

この例では次のメールボックス・データを提供します。

メールボックス・データ		属性番号	説明
割り付け	データ		
0x00	0x00	1	Offset = 0020h
0x01	0x20		サイズ = 16 バイト
0x02	0x00		
0x03	0x10		
0x04	0x00	2	Offset = 0050h
0x05	0x50		サイズ = 32 バイト
0x06	0x00		
0x07	0x20		
0x08	0x00	3	Offset = 0070h
0x09	0x70		サイズ = 32 バイト
0x0A	0x00		
0x0B	0x20		
0x0C	0x00	4	Offset = 00D0h
0x0D	0xD0		サイズ = 64 バイト
0x0E	0x00		
0x0F	0x40		

上記テーブルのように属性はマッピングされた順番で番号がつけられます。物理的にメールボックス・データのマッピング順を変更すると属性の番号も変更することができます。

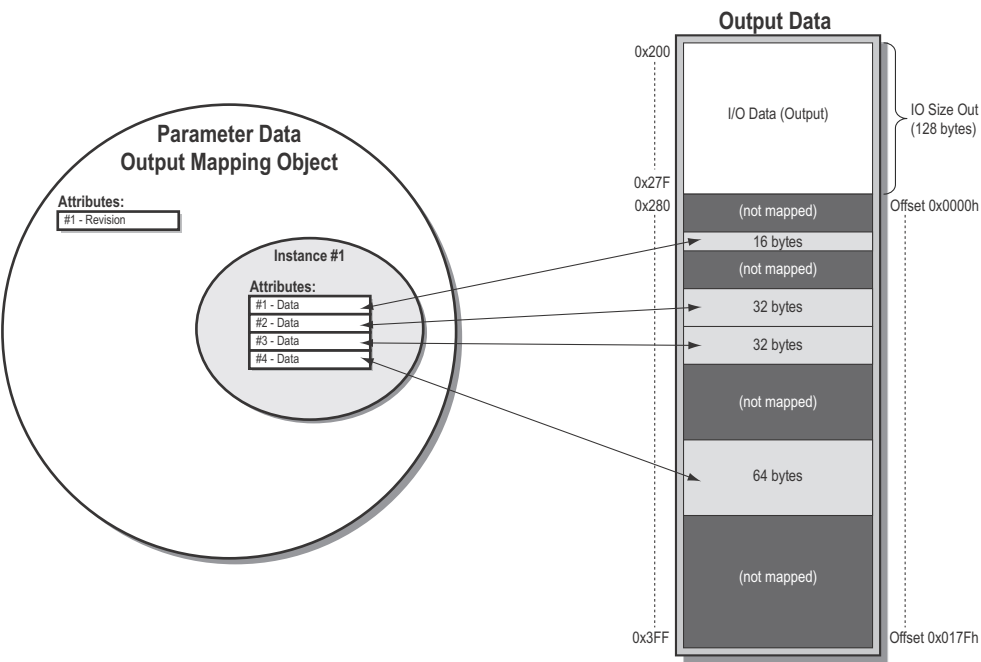
5. 新しいメールボックスを保存するために ‘File’ メニューの ‘Apply changes’ を選択します。

---

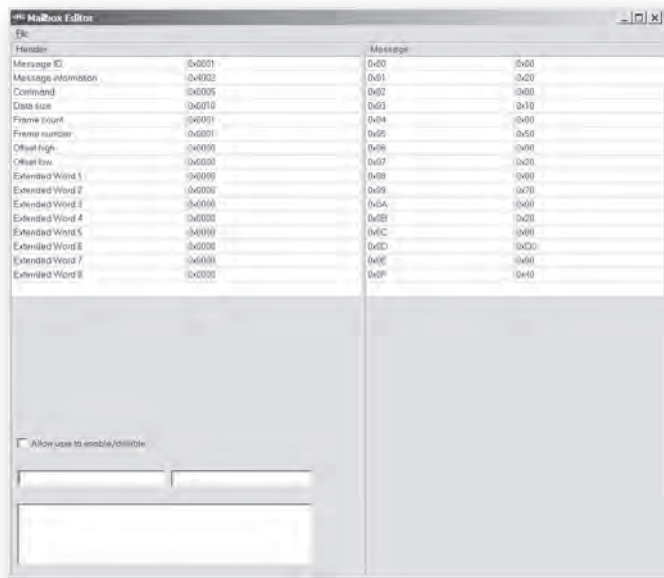
1. オフセットは ABC の物理メモリ割り付けではなく、パラメータ・データの開始から指定されます。



属性マッピングの結果



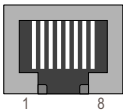
メールボックス・エディタ・スクリーンショット



# コネクタピン割り付け

## Ethernet コネクタ

ピン	信号
ハウジング	ケーブル・シールド
1	TD+
2	TD-
3	RD+
4	終端
5	終端
6	RD-
7	終端
8	終端



## 電源コネクタ

ピン	記述
1	+24V DC
2	GND

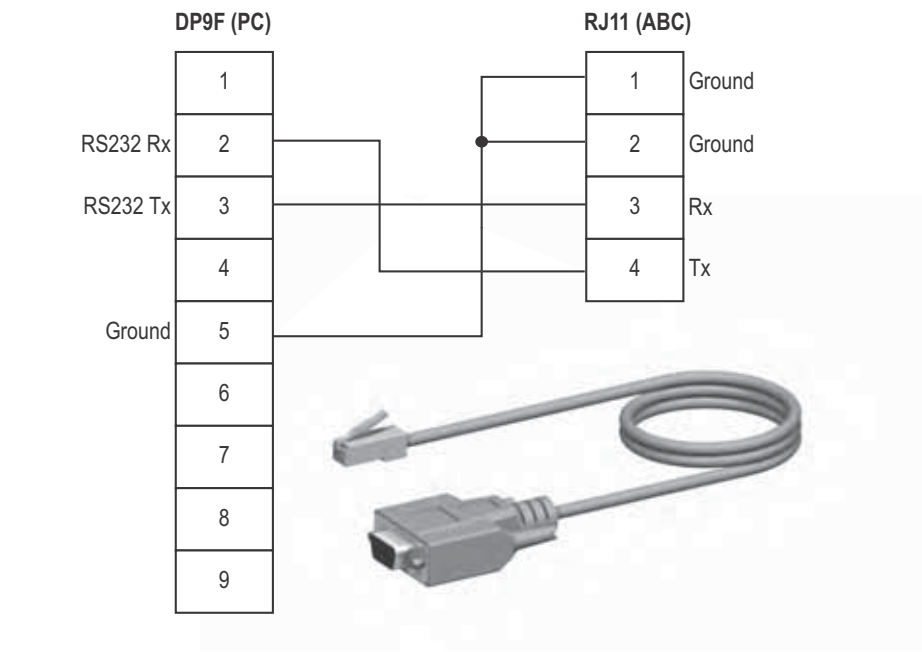


注意：

- 60/75 あるいは 75×C 銅線 (CU) のみ使用してください。
- ターミナル締め付けトルクは 5... 7 lbs-in (0.5... 0.8 Nm) 間でなければなりません。

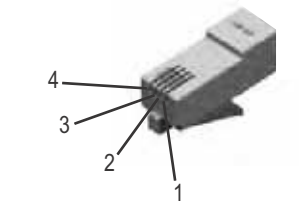
# PC コネクタ

## コンフィグレーション・ケーブル・ワイヤリング



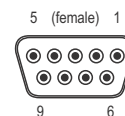
### RJ9 (ABC)

ピン	説明
1	シグナル・グラウンド
2	
3	RS232 Rx（入力）
4	RS232 Tx（出力）



**DB9F (PC)**

ピン	説明
1	-
2	RS232 Rx (入力)
3	RS232 Tx (出力)
4	-
5	シグナル・グラウンド
6-9	-



## サブネットワーク・インターフェース

### 概論

サブネットワーク・インターフェースは RS232、RS422 そして RS485 の通信用です。ABC コンフィグツールの設定次第で異なるシグナルがサブネットワーク・コネクタで有効になります。

### バイアス・レジスタ (RS485 のみ)

アイドル状態では RS485 は不定なステートになり、これによりシリアル・レシーバがシリアルラインからノイズを拾い、このノイズをデータとして解釈する可能性があります。これを避けるために、シリアル回線は一般的にバイアス・レジスタとして知られるプルアップ / プルダウン・レジスタを使用して一定のステートにしておく必要があります。

バイアス・レジスタは分圧器を形成し、差動ペア間の電圧がシリアル・レシーバの標準値 >200mV である閾値よりも高くなるように調節します。

バイアス・レジスタは1つのノードにのみ実装されることに注意してください。いくつものノードにバイアス・レジスタを実装するとネットワーク上のシグナルの質について妥協してしまい、伝送問題を引き起こすこともあります。

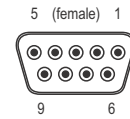
### 終端 (RS485&RS422 のみ)

終端のノード付近のシリアル・レシーバの間に終端レジスタを取り付け、サブネットワークを正しく終端させることが重要です。

抵抗値が標準 100...120R のケーブルの特性インピーダンスと一致することが理想的です。

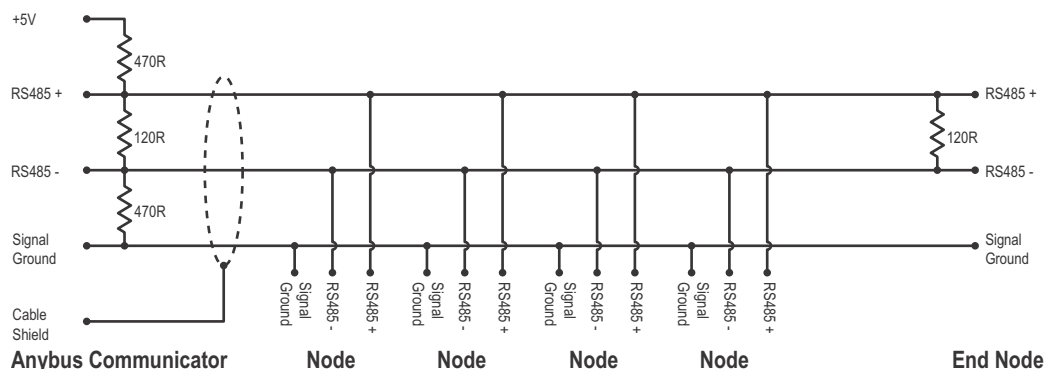
## コネクタ ピン配置 (DB9F)

ピン	説明	RS232	RS422	RS485
1	+5V 出力 (100mA 最大)	✓	✓	✓
2	RS232 Rx	✓		
3	RS232 Tx	✓		
4	(リザーブド)			
5	シグナル・グラウンド <sup>a</sup>	✓	✓	✓
6	RS422 Rx +		✓	
7	RS422 Rx -		✓	
8	RS485 + /RS422 Tx+		✓	✓
9	RS485 - /RS422 Tx-		✓	✓
(ハウジング)	ケーブル・シールド	✓	✓	✓

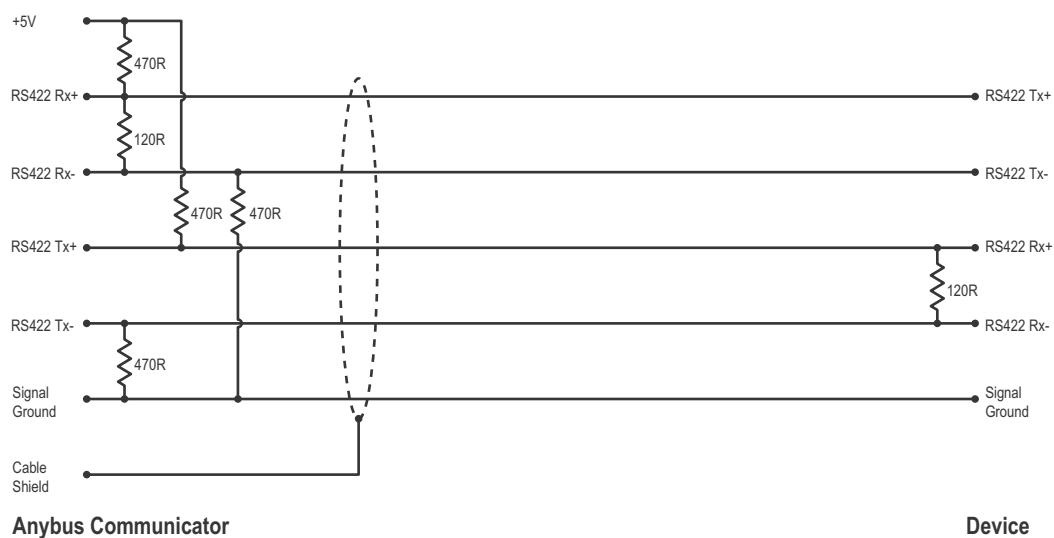


- a. グラウンド・ループなど、このシグナルを直接他のノードのプロテクティブ・アースに接続する場合、オンボード・シリアル・トランシーバにダメージを与える場合があります。一般的には他のノードのシグナル・グラウンド（利用可能であれば）への接続のみを推奨します。

## 代表的な接続 (RS485)

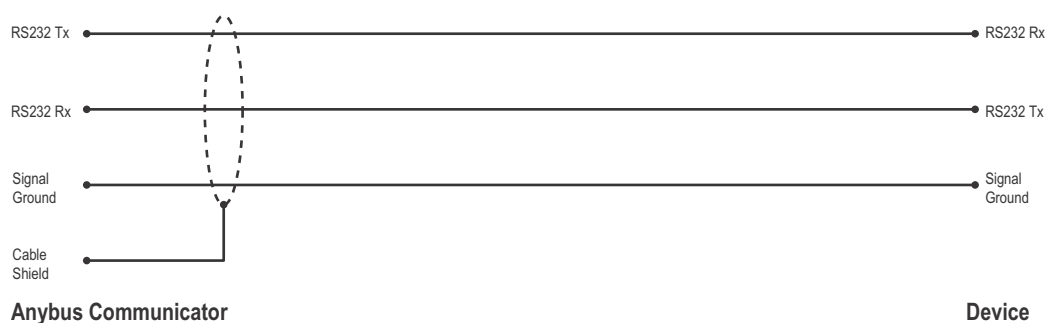


## 代表的な接続 (RS422 &amp; 4- ワイヤ RS485)



**Note:** Bias resistors are normally not needed on RS422, but may be required when using 4-wire RS485.

## 代表的な接続 (RS232)



## 技術仕様

### メカニカル仕様

#### 筐体

プラスチック筐体、スナップオンで DIN- レールに接続します。クラス IP20 のプロテクション。

#### サイズ

120 mm x 75 mm x 27 mm, L x W x H ( インチ : 4.72" x 2.95" x 1.06"; L x W x H)

### 電気的特性

#### 電源供給

電力 : 24V  $\pm$  10%

#### 消費電力

最大消費電流は 24V で 280mA です。通常は 100mA 前後になります。

### 環境特性

#### 相対湿度

製品は非凝結で 0 から 95% の相対湿度に対応しています。

#### 温度

動作中 :  $\pm 0^{\circ}\text{C}$  から  $+55^{\circ}\text{C}$

停止中 :  $-25^{\circ}\text{C}$  から  $+85^{\circ}\text{C}$

## 法的規制への遵守

### EMC 指令 (CE)

本製品は下記コンFORMANCEにより EMC 指令 89/336/EEC、92/31/EEC ならびに 93/68/EEC に準拠しています。

#### 試験内容

- EN 50082-2 (1993)

EN 55011 (1990)	Class A
-----------------	---------

- EN 61000-6-2 (1999)

EN 61000-4-3 (1996)	10V/m	
EN 61000-4-6 (1996)	10V/m	(all ports)
EN 61000-4-2 (1995)	±8kV	Air Discharge
	±4kV	Contact discharge
EN 61000-4-4 (1995)	±2kV	Power port
	±1kV	Other ports
EN 61000-4-5 (1995)	±0.5kV	Power ports (DM/CM)
	±1kV	Signal ports

### UL/c-UL 準拠

E214107 にて認証済み

### サブネットワーク・インターフェースの電氣的絶縁

- EN 60950-1 (2001)

#### 試験内容

Pollution Degree 2	
Material Group IIb	
250 V <sub>RMS</sub> or 250 VDC	Working voltage
500 V	Secondary circuit transient rating



## トラブルシューティング

問題	解決方法
<p>コンフィグレーション ダウンロード / アップロード中の問題。</p> <p>ABC コンフィグツールで Config Line "led" が赤になる。</p>	<ul style="list-style-type: none"> <li>シリアル通信がうまくいっていません。もう一度試してください。</li> </ul>
<p>シリアル・ポートは使用できるのにゲートウェイを接続することができません。</p>	<ul style="list-style-type: none"> <li>シリアル・ポートが他のアプリケーションで使用されている可能性があります。ABC コンフィグツールとシステム・トレイにあるものも含めてすべての他のアプリケーションを終了し、もう一度試してください。</li> <li>別のシリアル・ポートを選択し、もう一度試してください。</li> </ul>
<p>パフォーマンスが悪い</p>	<ul style="list-style-type: none"> <li>Navigation ウィンドウで 'Sub-Network' を右クリックしサブネットワークについてのステータス / 診断情報を見るために 'Sub-Network Status' を選択します。 もしゲートウェイが非常に多くの再伝送を報告していれば、ケーブルか、可能であればサブネットワーク向けにより低いボーレータを設定してみてください。</li> <li>ABC コンフィグツールの Sub-Net Monitor はアクティブであることを確認してください。 サブネット・モニタはゲートウェイ全体のパフォーマンスにネガティブな影響を与えます。必要なときのみ使用してください。</li> <li>ABC コンフィグツールの Node Monitor はアクティブか確認してください。 ノード・モニタはゲートウェイ全体のパフォーマンスにネガティブな影響を与えます。必要なときのみ使用してください。</li> </ul>
<p>サブネットワークが機能しない</p>	<ul style="list-style-type: none"> <li>サブネットワーク上のシリアルデータ通信を記録するためには 'Data logger' を使用してください。</li> <li>データが伝送されなければ、ABC コンフィグツールのコンフィグレーションをチェックしてください。</li> <li>データが受信されない場合、サブネットワーク・ケーブルをチェックしてください。また、伝送されたデータが正しいことを検証してください。</li> </ul>

# ASCII テーブル

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NUL 0	SOH 1	STX 2	ETX 3	EOT 4	ENQ 5	ACK 6	BEL 7	BS 8	HT 9	LF 10	VT 11	FF 12	CR 13	SO 14	SI 15
1x	DLE 16	DC1 17	DC2 18	DC3 19	DC4 20	NAK 21	SYN 22	ETB 23	CAN 24	EM 25	SUB 26	ESC 27	FS 28	GS 29	RS 30	US 31
2x	(sp) 32	! 33	" 34	# 35	\$ 36	% 37	& 38	' 39	( 40	) 41	* 42	+ 43	, 44	- 45	. 46	/ 47
3x	0 48	1 49	2 50	3 51	4 52	5 53	6 54	7 55	8 56	9 57	: 58	; 59	< 60	= 61	> 62	? 63
4x	@ 64	A 65	B 66	C 67	D 68	E 69	F 70	G 71	H 72	I 73	J 74	K 75	L 76	M 77	N 78	O 79
5x	P 80	Q 81	R 82	S 83	T 84	U 85	V 86	W 87	X 88	Y 89	Z 90	[ 91	\ 92	] 93	^ 94	_ 95
6x	` 96	a 97	b 98	c 99	d 100	e 101	f 102	g 103	h 104	i 105	j 106	k 107	l 108	m 109	n 110	o 111
7x	p 112	q 113	r 114	s 115	t 116	u 117	v 118	w 119	x 120	y 121	z 122	{ 123	 124	} 125	~ 126	DEL 127

---