


# Anybus<sup>®</sup> Communicator<sup>™</sup> for DeviceNet

User Manual

Doc. Id. JCM-1201-002

Rev. 2.52

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





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
	改定版リスト.....	P-2
	慣例と用語集.....	P-3
	.....	P-3
	用語集.....	P-3
	サポート.....	P-4
1 章	DeviceNet 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
2 章	基本操作	
	概論.....	2-1
	データ交換モデル.....	2-2
	メモリマップ.....	2-3
	データ交換例.....	2-4
	サブネットワーク・プロトコル.....	2-5
	プロトコル・モード.....	2-5
	プロトコル構築ブロック.....	2-5
	マスター・モード.....	2-6
	ジェネリック・データ・モード.....	2-7
	DeviceNet 上のデータ表示.....	2-8
	概論.....	2-8
	データ・タイプ.....	2-8
	メモリ・レイアウト.....	2-9
3 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.....	3-1
	プルダウン・メニュー.....	3-2
	ツールバー・アイコン.....	3-6

---

4 章	基本設定	
	フィールドバス設定 .....	4-1
	ABC パラメータ .....	4-1
	サブネットワーク・パラメータ .....	4-1
5 章	ノード	
	概論 .....	5-1
	ノードの追加と管理 5-1	
	ノード・パラメータ .....	5-1
6 章	トランザクション	
	概論 .....	6-1
	トランザクションの追加と管理 .....	6-2
	トランザクション・パラメータ (マスター・モード) .....	6-3
	パラメータ (クエリとブロードキャスト) .....	6-3
	パラメータ (レスポンス) .....	6-4
	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) .....	6-5
	プロデュース・トランザクション (Produce- Transactions) .....	6-5
	コンスーム・トランザクション (Consume- Transactions) .....	6-6
	トランザクション・エディタ .....	6-7
7 章	フレーム・オブジェクト	
	概論 .....	7-1
	フレーム・オブジェクトの追加と編集 .....	7-2
	通常オブジェクト (バイト、ワード、ダブルワード) .....	7-3
	制限オブジェクト (バイト、ワード、ダブルワード) .....	7-4
	データ・オブジェクト .....	7-5
	変数データ・オブジェクト .....	7-6
	チェックサム・オブジェクト .....	7-8
8 章	コマンド	
	概論 .....	8-1
	コマンドの追加と管理 .....	8-1
	プルダウン・メニュー .....	8-2
	ツールバー・アイコン .....	8-3
	コマンド・エディタ .....	8-3
	基本ナビゲーション .....	8-4
	プルダウン・メニュー .....	8-5
	コマンドの編集 .....	8-6
	例：マスター・モードでの Modbus-RTU コマンドの指定 .....	8-7

9 章	サブネットワーク・モニタ	
10 章	ノード・モニタ	
	概論 .....	10-1
	ノード・モニタのナビゲーション .....	10-2
	プルダウン・メニュー .....	10-3
	ツールバー・アイコン .....	10-4
11 章	データ・ロガー	
	概論 .....	11-1
	操作 .....	11-1
	コンフィグレーション .....	11-2
12 章	コンフィグレーション・ウィザード	
	概論 .....	12-1
	ウィザード・プロファイルの選択 .....	12-1
	ウィザード - Modbus RTU マスター .....	12-2
13 章	制御 / ステータス・レジスタ	
	概論 .....	13-1
	ハンドシェーク手順 .....	13-2
	データの一貫性 .....	13-2
	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) .....	13-4
	概論 .....	13-4
	マスター・モードでのステータス・コード .....	13-5
	ジェネリック・データ・モードでのステータス・コード .....	13-5
	制御レジスタ構成 (制御システムからゲートウェイへ) .....	13-6
	概論 .....	13-6
	マスター・モードでの制御コード .....	13-6
	ジェネリック・データ・モードでの制御コード .....	13-6
14 章	CIP オブジェクトの実装	
	識別オブジェクト、クラス 01h .....	14-2
	概論 .....	14-2
	クラス属性 .....	14-2
	インスタンス属性 .....	14-2
	メッセージ・ルータ、クラス 02h .....	14-4
	概論 .....	14-4
	クラス属性 .....	14-4
	インスタンス属性 .....	14-4

---

DeviceNet オブジェクト、クラス 03h.....	14-5
概論 .....	14-5
クラス属性 .....	14-5
インスタンス属性 .....	14-5
アセンブリ・オブジェクト、クラス 04h.....	14-6
概論 .....	14-6
クラス属性 .....	14-6
インスタンス属性 – インスタンス / 接続ポイント 64h .....	14-6
インスタンス属性 – インスタンス / 接続ポイント 96h .....	14-6
接続オブジェクト、クラス 05h.....	14-7
概論 .....	14-7
クラス属性 .....	14-7
インスタンス 1 & 10..14 (エクスプリシット・メッセージング接続) 属性.....	14-7
インスタンス 2 (ポーリングされた接続) 属性.....	14-8
インスタンス 3 (ビット・ストローブ接続) 属性 .....	14-8
インスタンス 4 (COS/ サイクリック接続) 属性.....	14-8
承認ハンドラ・オブジェクト、クラス 2Bh.....	14-10
概論 .....	14-10
クラス属性.....	14-10
インスタンス属性 .....	14-10
診断オブジェクト、クラス AAh.....	14-11
概論 .....	14-11
クラス属性 .....	14-11
インスタンス属性、インスタンス 01h .....	14-11
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.....	14-12
概論 .....	14-12
クラス属性 .....	14-12
インスタンス属性、インスタンス 01h .....	14-12
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.....	14-13
概論 .....	14-13
クラス属性 .....	14-13
インスタンス属性、インスタンス 01h .....	14-13

15 章	先進フィールドバス・コンフィグレーション	
	概論.....	15-1
	メールボックス・エディタ.....	15-2
アペンディックスA	パラメータ・データの初期化 (エクスプリシット・データ)	
	概要.....	A-1
	メールボックス・メッセージの追加.....	A-1
	DeviceNet への入力パラメータ・データのマッピング.....	A-2
	DeviceNet への出力パラメータ・データのマッピング.....	A-5
アペンディックスB	コネクタ ピン割り付け	
	DeviceNet Connector コネクタ.....	B-1
	電源コネクタ.....	B-1
	PC コネクタ.....	B-1
	サブネットワーク・インターフェース.....	B-2
	概論.....	B-2
	バイアス・レジスタ (RS485 のみ).....	B-2
	終端 (RS485&RS422 のみ).....	B-2
	コネクタ ピン配置 (DB9F).....	B-3
	代表的な接続 (RS485).....	B-3
	代表的な接続 (RS422 & 4-ワイヤ RS485).....	B-4
	代表的な接続 (RS232).....	B-4
アペンディックスC	技術仕様	
	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
アペンディックスD	トラブルシューティング	
アペンディックスE	ASCII テーブル	

## 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はみなされません。

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。

警告	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。
ESD ノート	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



## 関連マニュアル

マニュアル名	製作者
ABC-DEV Installation Leaflet	HMS

## マニュアル更新履歴

### 最近の更新 (2.51…2.52)

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	4-1
"On data change" パラメータ動作の更新	6-2 - 6-4

### 改定版リスト

改定番号	改定日	作成者	章	説明
2.00	2003-12-03	PeP	全章	2 回目のメジャー・リリース制御コードの訂正
2.01	2004-03-08	PeP	第 15 章	制御コードの訂正
2.50	2006-04-05	PeP	全章	大幅な書き直し
2.51	2006-12-22	PeP	-	その他マイナー修正と更新
2.52	2009-04-23	KeL	全章	その他マイナー修正と更新

## 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® Communicator™
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
DEV	DeviceNet
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、DeviceNet
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 (<http://www.anybus.jp>) 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: <http://www.hms.se>

### 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: <http://www.anybus.de>

### 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: <http://www.anybus.jp>

### HMS China

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

### 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: <http://www.anybus.it>

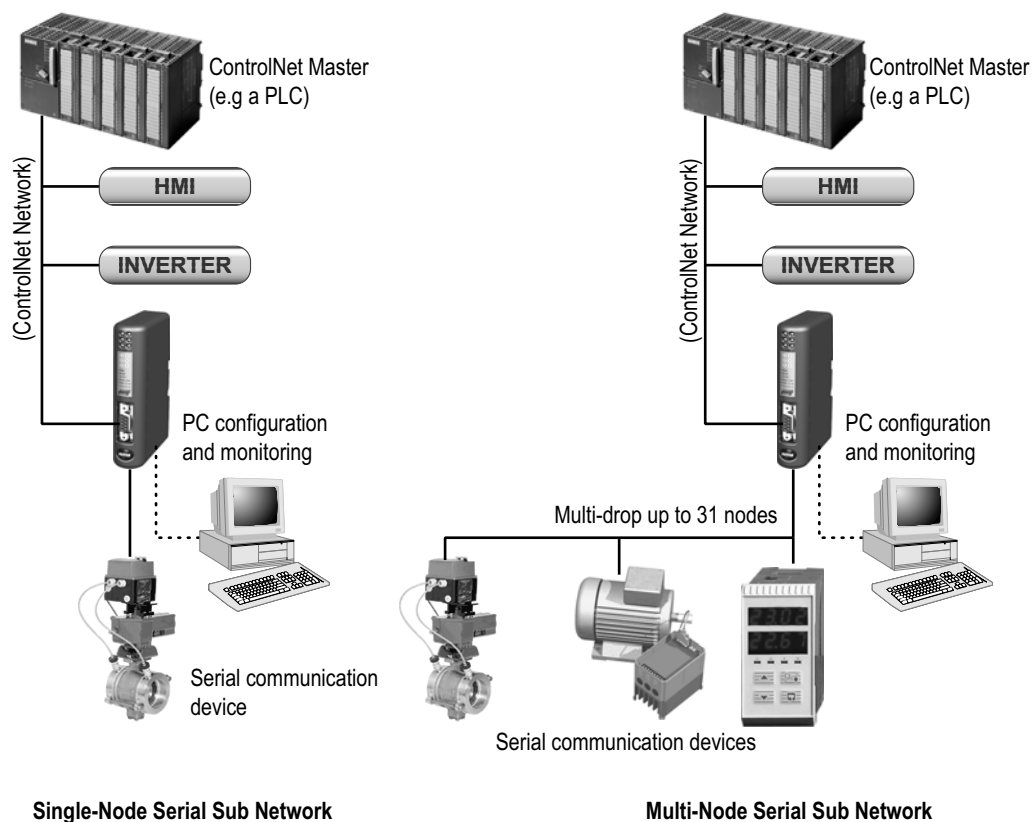
### HMS France

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: <http://www.anybus.fr>

## DeviceNet 向け Anybus Communicator について

DeviceNet 向け Anybus Communicator は、シリアル・アプリケーション・プロトコルと DeviceNet 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

### DeviceNet インターフェース

DeviceNet 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。

- 通信アダプタ、プロファイル No.12
- グループ 2 のみサーバ
- オン・ボード・スイッチを介して MacID とボーレートのコンフィグレーション
- ポーリング、ステートの変更、そしてビットでストロボされた I/O

## 外観

結線とピン配置

参照 ⇒ “コネクタ ピン割り付け” B-1

### A: DeviceNet 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。

参照 ⇒ “DeviceNet Connector コネクタ” B-1

### B: コンフィグレーション・スイッチ

参照 ⇒ “コンフィグレーション・スイッチ” 1-4

### C: LED ステータス

参照 ⇒ “LED ステータス” 1-3

### 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。

参照 ⇒ “PC コネクタ” B-1

### 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。

参照 ⇒ “DeviceNet Connector コネクタ” B-1

### F: 電源コネクタ

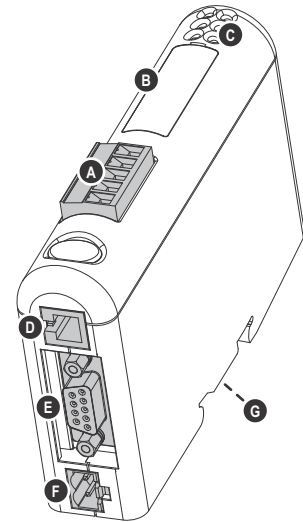
このコネクタは電源を ABC に供給するために使用します。

参照 ⇒ “電源コネクタ” B-1 “メカニカル仕様” C-1

### 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。

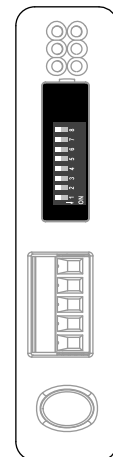
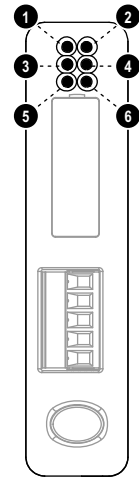
参照 ⇒ “メカニカル仕様” C-1



## LED ステータス

#	ステート	ステータス
1- ネットワーク・ステータスオフ	オフ	電源が供給されていない / オフライン
	グリーン	リンク OK、オンライン、接続
	グリーン、点滅	オンライン、非接続
	レッド	クリティカルなリンク失敗
	レッド、点滅	接続タイムアウト
2- モジュール・ステータス	オフ	デバイスに電源が与えられていない
	グリーン	デバイス動作中
	グリーン、点滅	データサイズがコンフィグレーションより大きい
	レッド	復旧できないフォルト
	レッド、点滅	マイナーフォルト
3- 使用	-	-
4- 未使用	-	-
5- サブネット・ステータス <sup>a</sup>	オフ	電源オフ
	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	電源オフ
	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	機器のステータス LED が一度あるいはそれ以上のレッド点滅を続ける場合、シーケンス・パターンに注意してください。HMS サポートまでご連絡ください。

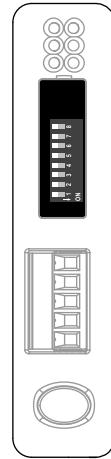
- a. この LED は全てのトランザクションが最低一度でもアクティブになったとき緑に変わります。これは "change of state" または "change of state on trigger" を使用したどんなトランザクションも含まれます。トランザクションでタイムアウトが発生した場合は、この LED は赤に変わります。



## コンフィグレーション・スイッチ

コンフィグレーション・スイッチは DeviceNet MacID とボーレートの設定をするために使用されます。

**注意！** これらの設定は動作中に変更することはできません。ABC は変更を反映させるためにリセットが必要です。



スイッチは次のように解釈されます。

ボーレート	sw.1	sw.2
125k	OFF	OFF
250K	OFF	ON
500K	ON	OFF
(リザーブド)	ON	ON

Mac ID	sw.3	sw.4	sw.5	sw.6	sw.7	sw.8
0	OFF	OFF	OFF	OFF	OFF	OFF
1	OFF	OFF	OFF	OFF	OFF	ON
...	...	...	...	...	...	...
62	ON	ON	ON	ON	ON	OFF
63	ON	ON	ON	ON	ON	ON

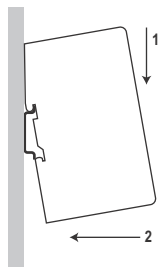
## ハードウェアのインストール

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。

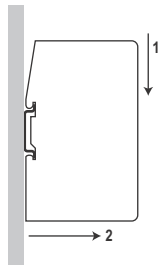
1. ABC を DIN レールにパチッと音がするまで取り付けます。

参照 ⇨ “G: DIN レール・コネクタ” 1-2

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 DeviceNet 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。
7. PC で ABC コンフィグツールをスタートします。  
(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。)
8. ABC コンフィグツールを使用して ABC をコンフィグレーションしコンフィグレーションをダウンロードします。



## 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。(http://www.anybus.jp)

### 電子データシート (EDS ファイル)

DeviceNet 上で、デバイスの特性は EDS が拡張子の ASCII データ・ファイルに保存されます。このファイルはネットワークの設定のときに DeviceNet のコンフィグレーションツールによって使用されます。このファイルの最新バージョンは HMS ウェブサイト 'http://www.anybus.jp' から入手できます。

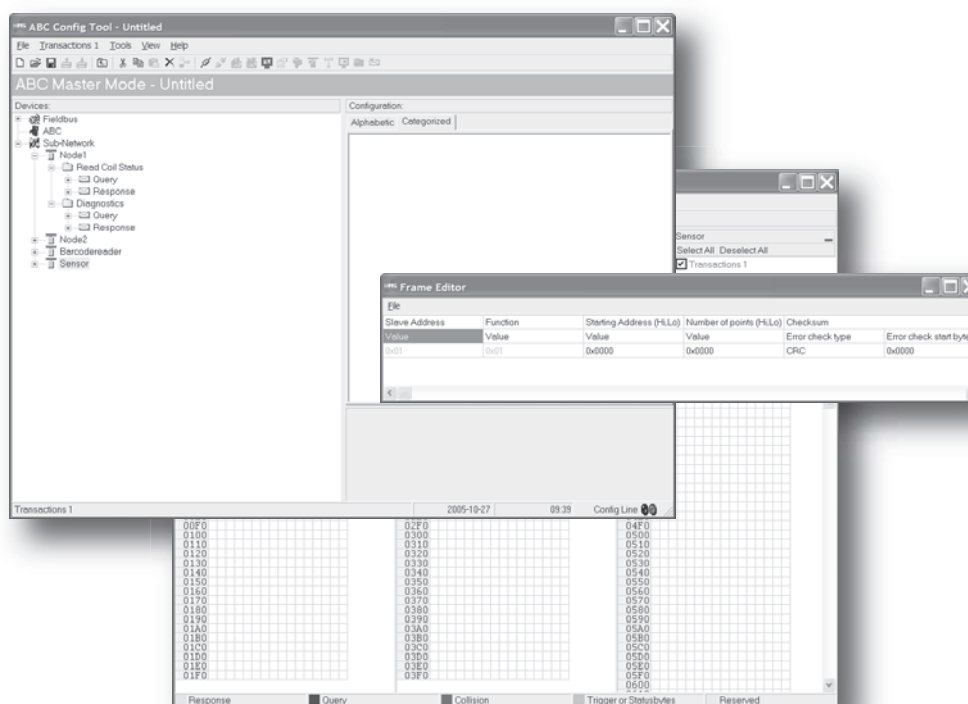
# 基本操作

## 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。



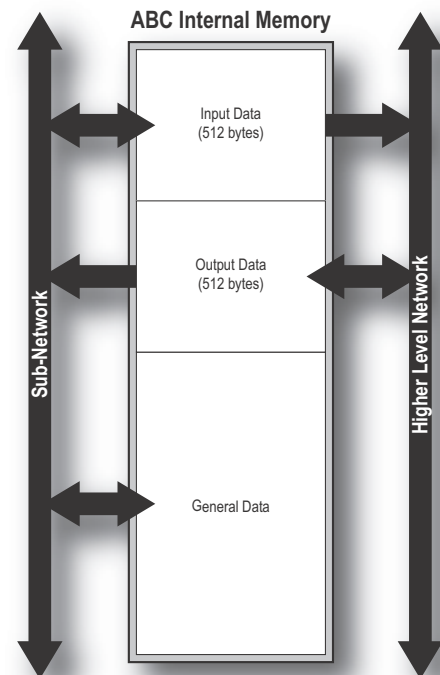
## データ交換モデル

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。

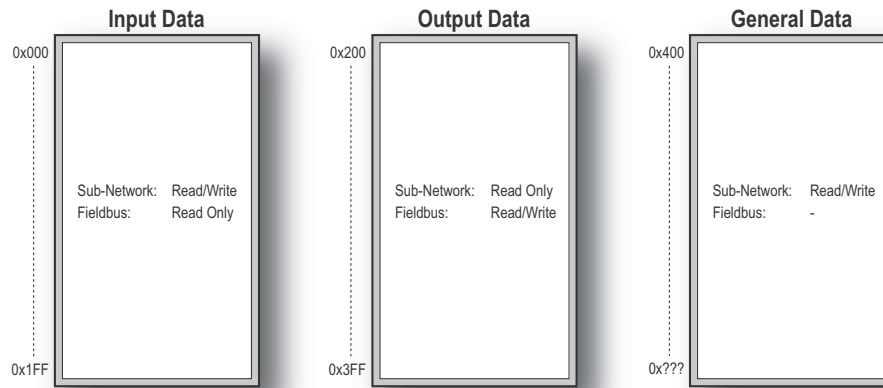
内部メモリ・バッファは機能に基づいて3つの領域に分けられます。

- 入力データ (512 バイトまで)  
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 (512 バイトまで)  
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ず。)
- 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。



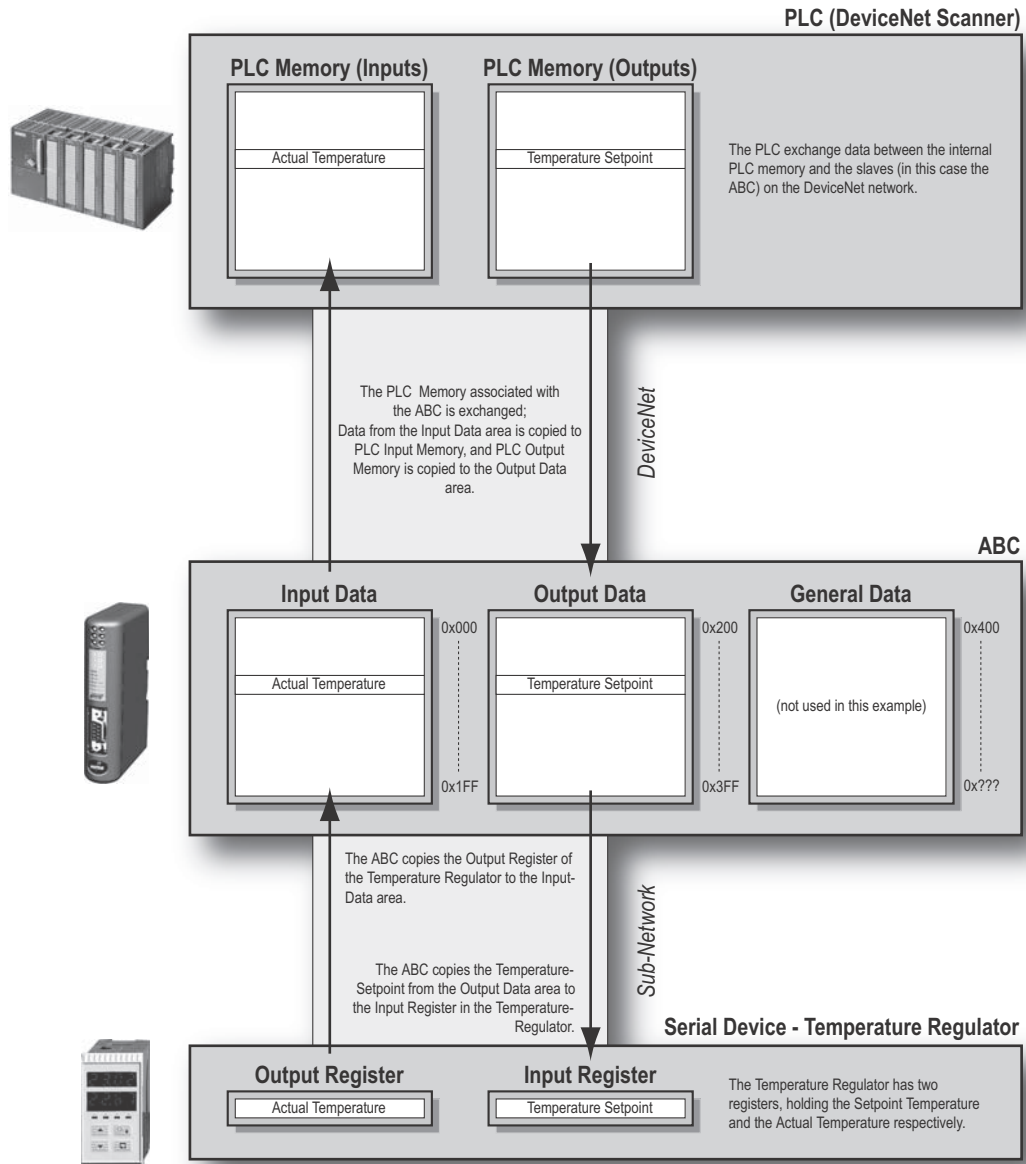
## メモリマップ

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。

**注意！** このプロトコルモードは、実際のサブネットワークプロトコルではなく、ベシクコミュニケーションのみを指します。

- 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 / レスポンズ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。

参照 → “ジェネリック・データ・モード” 2-7

### 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。

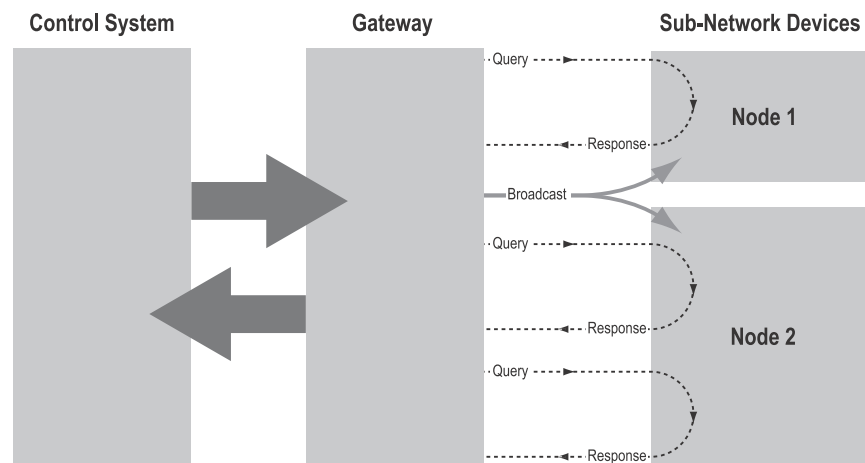
- フレーム・オブジェクト

'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。

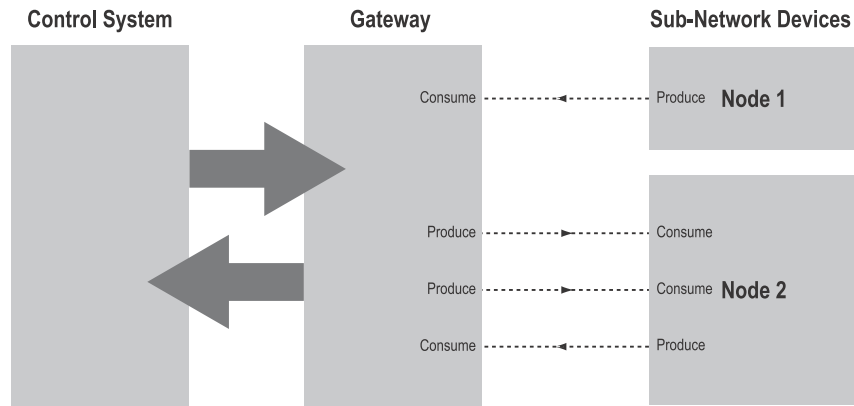
このルールには一つ例外があり、それがブロードキャストです。ほとんどのプロトコルはネットワーク上の全てのノードに対して応答を必要としないブロードキャストメッセージを送信する方法を提供します。これはブロードキャスト・ノードと通信する機能がある ABC でも使用されます。



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sums します。ノードはメッセージに回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sums' しています。この 'コンsums' 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sums' されるメッセージの 'プロデュース' に使用されます。



## DeviceNet 上のデータ表示

### 概論

DeviceNet 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。

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。

参照⇒ “CIP オブジェクトの実装” 14-1

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

参照⇒ “アセンブリ・オブジェクト、クラス 04h” 14-6

“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” 14-12

“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” 14-13

“基本設定” 4-1

## メモリ・レイアウト

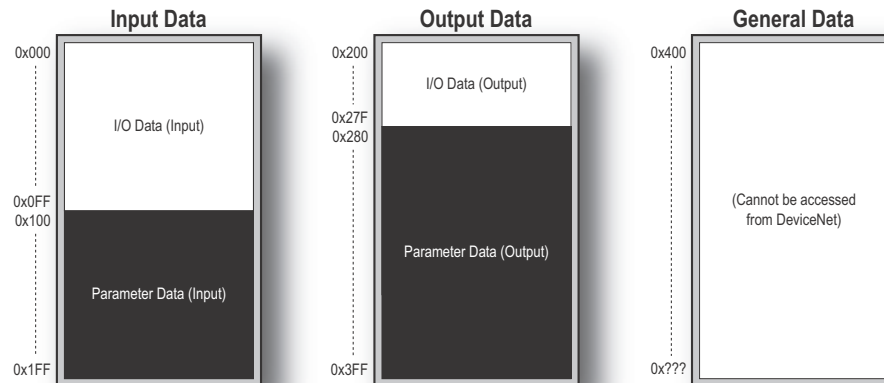
入力 / 出力データ領域のデータは次のように継続的な I/O データ・ブロックとして表示されます。

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)

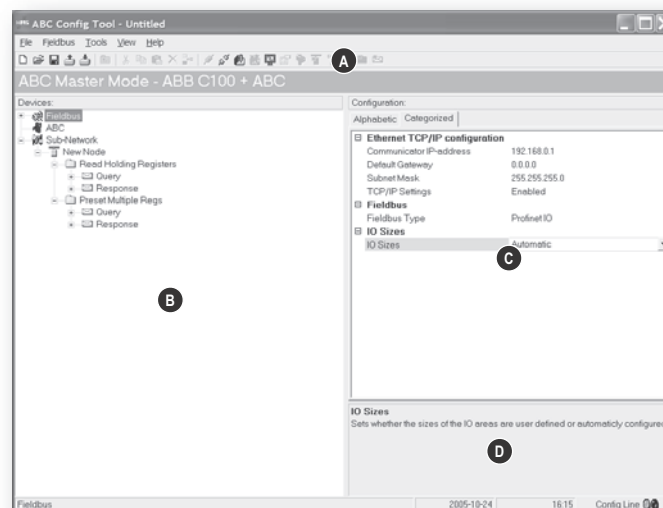
メモリ・レイアウトの結果



## 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

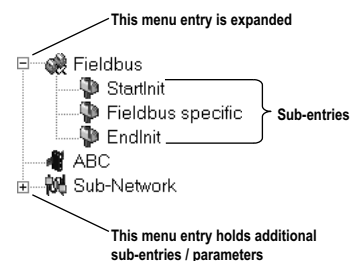
左側の二番目のドロップダウン・メニューはそのときの選択で変わります。ツールバーによって最もよく使う機能に速くアクセスできます。

- 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。



Configuration	
Alphabetic	Categorized
[-] Communication	
Baudrate (bits/s)	9600
Data bits	8
Parity	None
Physical standard	RS232C
Start bits	1
Stop bits	1
[-] Timing	
Message delimiter (10ms)	0

Parameter Section

Configuration	
Alphabetic	Categorized
[-] Timing	
Message delimiter (10ms)	0

Information Section

- 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。

## 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

参照 ⇒ “コンフィグレーション・ウィザード” 12-1

- 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

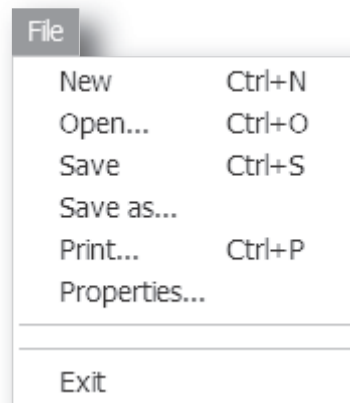
新しい名前で現在のコンフィグレーションを保存します。

- Print...

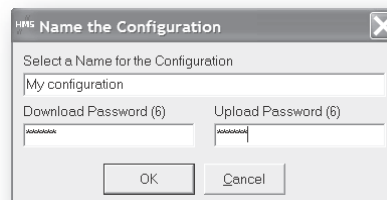
現在のコンフィグレーションの詳細を印刷します。

- 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

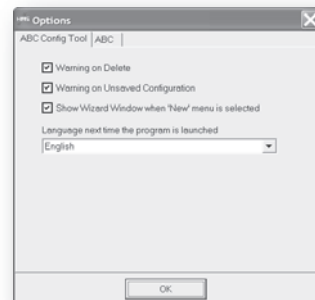
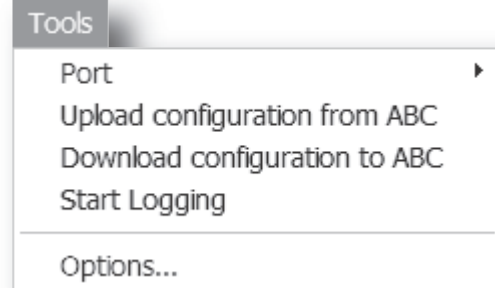
参照 ⇨ “データ・ロガー” 11-1

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...

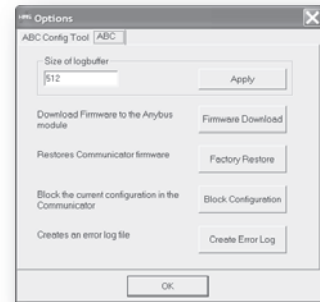
このメニューは次のウィンドウを表示します。

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	<p>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）。</p> <p>'Apply' をクリックし新しい設定を有効にしてください。</p> <p><b>参照</b> ⇨ “データ・ロガー” 11-1”</p>
Firmware Download	<p>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。</p> <p><b>注意</b> ご使用には注意してください。</p>
Factory Restore	<p>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）。</p>
Block Configuration	<p>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。</p> <p><b>注意</b> ご使用には注意してください。</p>
Create Error log	<p>エラー・ログファイルを作成します。</p>



## View

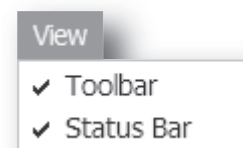
このメニューには次のエントリがあります。

- 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。

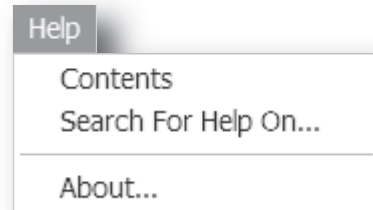
- Search For Help On...

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

参照 ⇨ “File” 3-2



- Upload from ABC & Download to ABC

参照 ⇨ “Tools” 3-3.



- 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

参照 ⇨ “Tools” 3-3、“データ・ロガー” 11-1



- Sub-Network Monitor

このアイコンをクリックし、Sub-network Monitor を起動します。

参照 ⇨ “サブネットワーク・モニタ” 9-1



- 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。



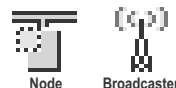
- Add Mailbox

参照 ⇨ メールボックス・メッセージ機能について：“メールボックス・エディタ” 15-2



- 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。

参照 ⇨ “ノード・モニタ” 10-1





- 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)。

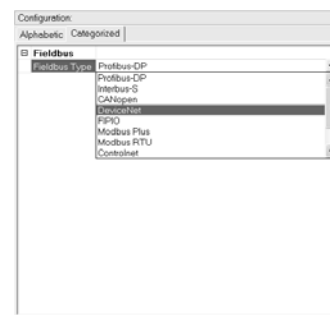


#### 概論

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 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

#### 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。

- 自動

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

- ユーザ定義

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。



IO Sizes

参照 ⇨ “DeviceNet 上のデータ表示” 2-8

“アセンブリ・オブジェクト、クラス 04h” 14-6

“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” 14-12

“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” 14-13

## 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

参照 ⇨ “制御 / ステータス・レジスタ” 13-1

値	説明
Enabled	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。

参照 ⇨ “Protocol Mode” 4-2

### 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 10  $\mu$ s 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

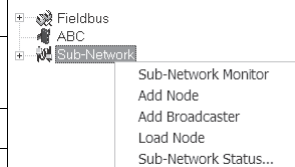
## 概論

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

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 (9-1 "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。  参照 ⇨ “コマンド・エディタ” 8-3



# 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

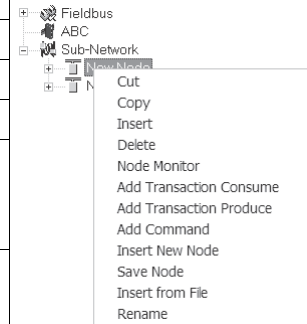
トランザクションは双方向に追加することができます。サブネットワークへ送信されたトランザクションは 'Transaction Produce' と呼ばれ、他のノードにより発行されたトランザクションは 'Transaction Consume' と呼ばれます。

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	Delete a node ノードを削除します。
Node Monitor	ノード・モニタの起動をします。 参照 ⇨ “ノード・モニタ” 10-1
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	'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)	<p>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。</p> <p><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。</p>
Offline options for fieldbus	<p>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。</p> <ul style="list-style-type: none"> <li>• <b>Clear</b> 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>• <b>Freeze</b> スレーブ機器へのデータがフリーズされます。</li> <li>• <b>NoScanning</b> サブネットワークの更新が停止します。</li> </ul>
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。</p> <ul style="list-style-type: none"> <li>• <b>Clear</b> 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>• <b>Freeze</b> 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)	<p>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。</p> <p><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。</p>
Retries	<p>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。</p>
Timeout time (10ms)	<p>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。</p>
Trigger byte address	<p>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)</p>



パラメータ	説明
Update mode	<p>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。</p> <ul style="list-style-type: none"> <li>• <b>Cyclically</b> ‘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>• <b>On data change</b> 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>• <b>Single shot</b> スタート時にクエリが送信されます。</li> <li>• <b>Change of state on trigger</b> 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	<p>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。</p> <p>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。</p>
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>

## 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 field-bus	<p>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。</p> <ul style="list-style-type: none"> <li>• <b>Clear</b> 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>• <b>Freeze</b> 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>• <b>NoScanning</b> 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	<p>トランザクションの更新モード：</p> <ul style="list-style-type: none"> <li>• <b>Cyclically</b> 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>• <b>On data change</b> 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>• <b>Single shot</b> 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>• <b>Change of state on trigger</b> 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>
Update time (10ms)	<p>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)。</p>

パラメータ	説明
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>

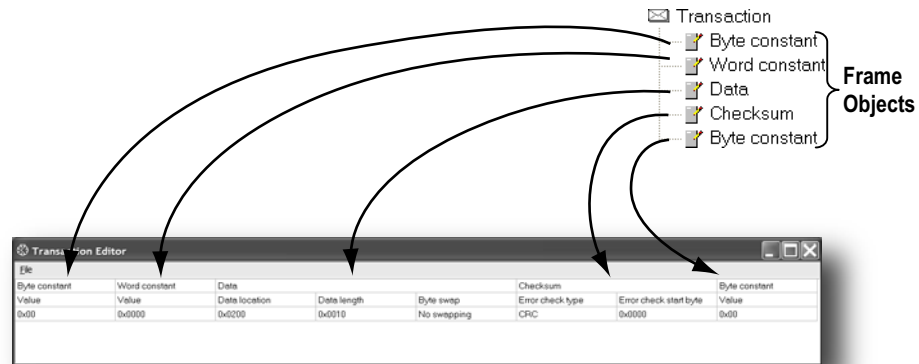
## コンスーム・トランザクション (Consume- Transactions)

(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せん。</p> <ul style="list-style-type: none"> <li>• <b>Clear</b> 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>• <b>Freeze</b> 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)	<p>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。</p>
Trigger byte	<ul style="list-style-type: none"> <li>• <b>Enable</b> 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>• <b>Disable</b> 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>

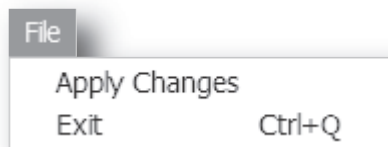
## トランザクション・エディタ

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。



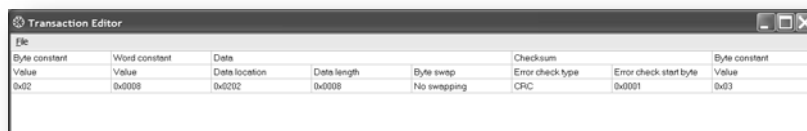
パラメータの値の編集をするためクリックしキーボードを使用して新しい値を入力してください。既に設定されたコマンドを使用してトランザクションの編集をしているとき、トランザクションによっては編集できないかもしれません。

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 0x202 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

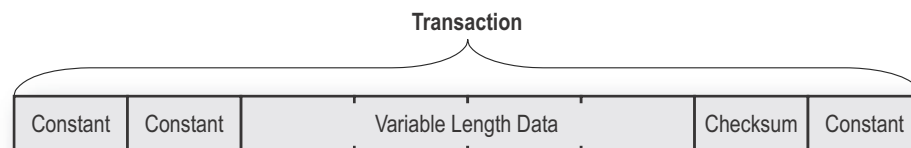
各トランザクションはシリアル・テレグラム・フレームを生成するフレーム・オブジェクトを含みます。各フレーム・オブジェクトはゲートウェイにテレグラムの特定の部分の処理方法または生成方法を指定します。

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

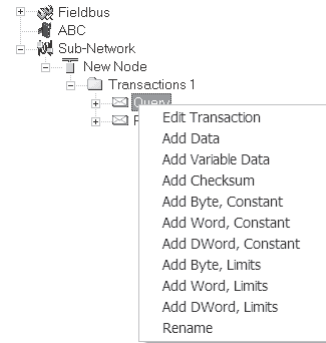
例：

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。



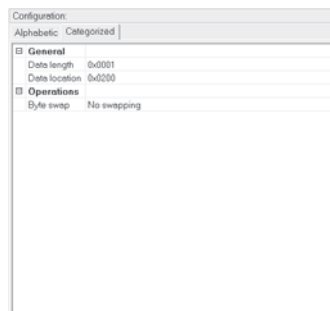
## フレーム・オブジェクトの追加と編集

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。

参照 ⇨ “トランザクション・エディタ” 6-7



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。

参照 ⇨ “トランザクション・エディタ” 6-7

## 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
- ワード  
16 ビット
- ダブルワード  
32 ビット

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。

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>Range:0x00... 0xFFh (バイト) 0x0000... 0xFFFFh (ワード) 0x00000000... 0xFFFFFFFFh (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。(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>Range:0x00... 0xFEh (バイト) 0x0000... 0xFFFEh (ワード) 0x00000000... 0xFFFFFEEh (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。

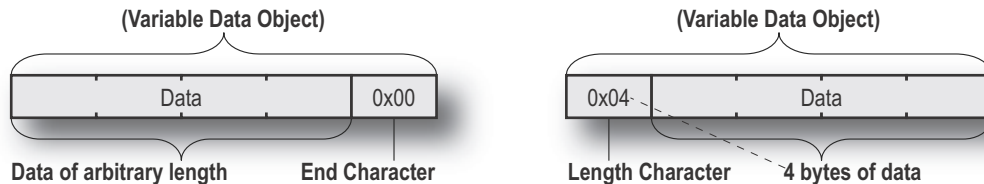
オブジェクトのプロパティを指定するため、Navigation セクション内でこれを選択し、次のように Parameter セクションで設定する値を入力します。

パラメータ	説明
Byte Swapping	<ul style="list-style-type: none"> <li>• <b>No Swapping</b> データのスワップはされません。</li> <li>• <b>Swap 2 bytes</b> A, B, C, D が B, A, D, C になります。</li> <li>• <b>Swap 4 bytes</b> 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>• <b>No Swapping</b> データのスワップはされません。</li> <li>• <b>Swap 2 bytes</b> A, B, C, D が B, A, D, C になります。</li> <li>• <b>Swap 4 bytes</b> 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>• <b>Enabled<sup>a</sup></b> '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>• <b>Disabled</b> 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

パラメータ	説明
Object Delimiter	<ul style="list-style-type: none"> <li>• <b>Length Character</b> 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ができません。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>• <b>End Character</b> 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ができません。</li> <li>• <b>End Character Visible</b> 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>• <b>No Character<sup>a</sup></b> 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 Only relevant for Consume/Response transactions
- 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の計算を開始するバイト・オフセットを指定します。
Error Check Type	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。 <ul style="list-style-type: none"><li>• <b>CRC (2 bytes)</b> 0xFFFF による CRC-16 多項式 (Modbus RTU 標準)</li><li>• <b>LRC (1 byte)</b> 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>• <b>XOR (1 byte)</b> 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>• <b>ADD (1 byte)</b> 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>• <b>AddInvASCII (2 bytes)</b> 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。

参照 ⇨ “ノード・パラメータ” 5-1

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。

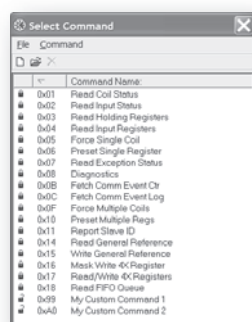
マスター・モードでは、ABC コンフィグツールは最も一般的な Modbus RTU 機能を使用し予めロードされます。コマンドの追加はコマンド・エディタを使用して簡単に行えます。

参照 ⇨ “コマンド・エディタ” 8-3

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

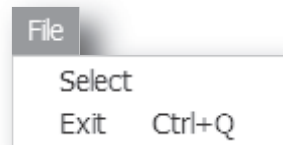
このメニューには次のエントリがあります。

- **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。

参照 ⇨ “コマンド・エディタ” 8-3

- **Edit Command**

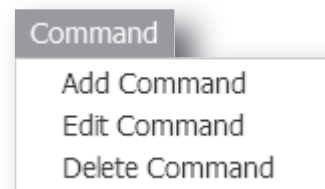
コマンド・エディタを使用して現在選択されているコマンドを編集します。

参照 ⇨ “コマンド・エディタ” 8-3

- **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。

**注意！** 削除できないコマンドもあります。



## ツールバー・アイコン

ツールバーには最も一般的に使用される機能のアイコンがあります。

- **Add Command**

参照 ⇨ “Add Command” 8-2



- **Edit Command**

参照 ⇨ “Edit Command” 8-2



- **Delete Command**

参照 ⇨ “Delete Command” 8-2



## コマンド・エディタ

### 概論

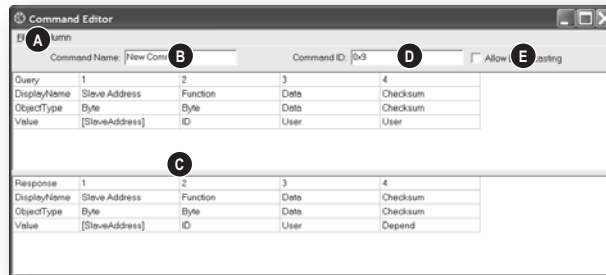
コマンド・エディタは新しいコマンドを設定し既存のコマンドを編集するために使用します。この機能により後に再利用することができるコマンドのライブラリを作成することができます。

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' を選び、コマンド・エディタを開きます。



### A: プルダウン・メニュー

参照 ⇒ “プルダウン・メニュー” 8-2

### 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	コマンドは消費されるデータです。(ジェネリック・データ・モードのみ)

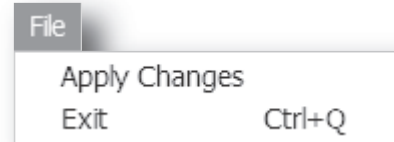


## プルダウン・メニュー

### File

このメニューには次のエントリがあります。

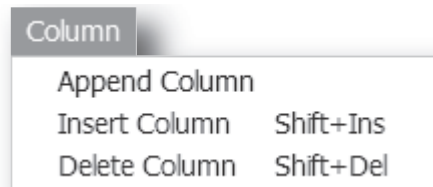
- **Apply Changes**  
変更を保存してメイン・ウィンドウに戻ります。
- **Exit**  
保存せずにメイン・ウィンドウに戻ります。



### 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。

値	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。 参照 ⇨ “基本ナビゲーション” 8-4
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。 参照 ⇨ “ノード・パラメータ” 5-1
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を使用されます。

- 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
Display-Name	Slave Address	Function	Data	Checksum
Object Type	Byte Object	Byte Object	Data Object	Checksum Object
Value	[SlaveAddress] この Byte constant の値は 'SlaveAddress' パラメータを使用し設定されます。 参照 ⇨ “ノード・パラメータ” 5-1	ID この Byte constant の値は 'Command ID' フィールドを使用し設定されます。	User 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。	User 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
Display-Name	Slave Address	Function	Data	Checksum
Object Type	Byte Object	Byte Object	Data Object	Checksum Object
Value	[SlaveAddress] この値は Parameter Window の 'SlaveAddress' パラメータにリンクします。	ID Byte Constant の値は 'Command ID' フィールドを使用し設定されます。	User 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。	Depend 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

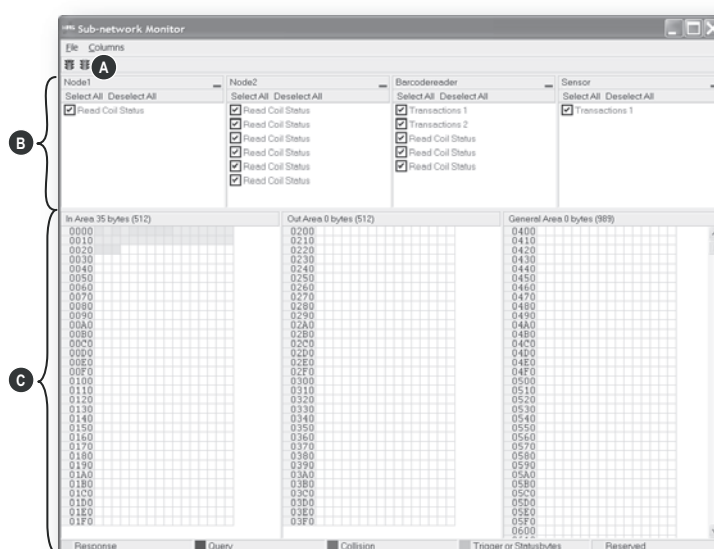
サブネットワーク・モニタはサブネットワークのコンフィグレーションとトラブルシューティングを簡単にします。

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。



### 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

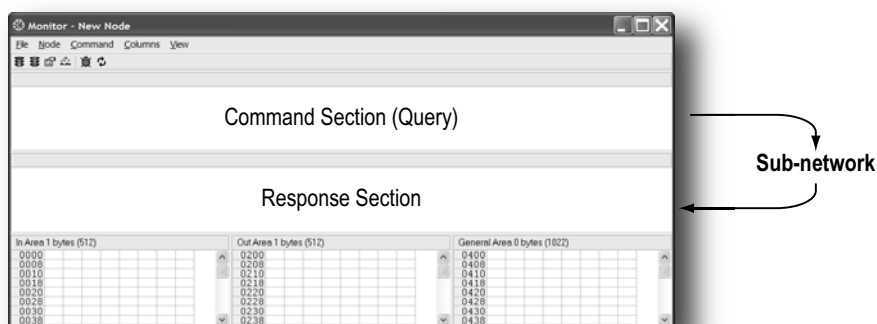
手動で個別のコマンドを送信できる設定を行い（使用できる場合）レスポンスをモニタすることでノード・モニタはサブネットワークの通信の設定のための情報を提供します。特定のノードによって使用されているメモリの概要も提供します。

**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。

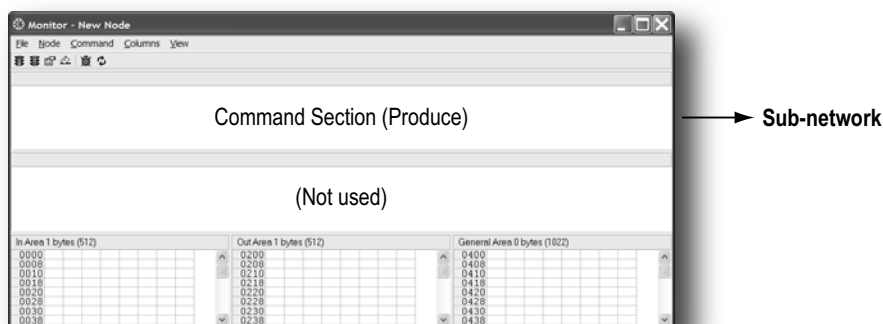
### ・ マスター・モード

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。

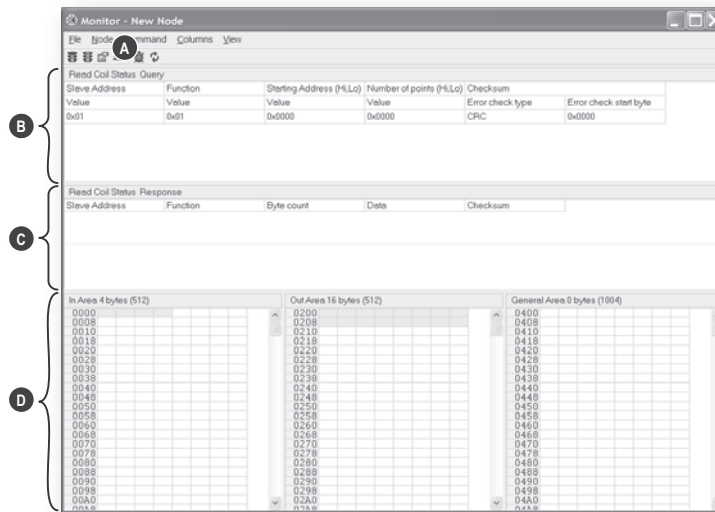


### ・ ジェネリック・データ・モード

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**

参照 ⇒ “プルダウン・メニュー” 10-3 、 “ツールバー・アイコン” 10-4

**B: コマンド・セクション Command Section**

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) Section (Master Mode only)**

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 Monitor Section**

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示します。

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。

参照 ⇒ “ツールバー・アイコン” 10-4

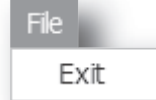
## プルダウン・メニュー

### 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

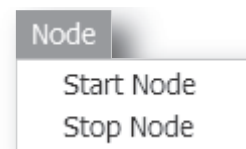
このメニューはノードのデータ交換を制御します。特定のノードに関連した問題を切り分けるために使用できます。

- **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

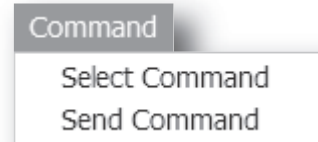
このメニューは手動でコマンドを指定し送信するために使用します。

- **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。



### Column

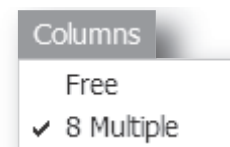
このメニューは Monitor セクション内のカラム数を指定します。

- **Free**

カラム数はウィンドウの幅に依存します。

- **8 Multiple**

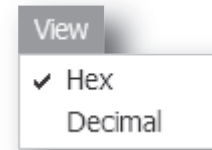
カラム数は 8 に固定されます。





## 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 This menu specifies the data representation in the Monitor Section.



- **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、一般的に使用頻度の高い機能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。

参照 ⇨ “Node” 10-3



### • 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。

参照 ⇨ “Command” 10-3



### • 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）。



### • Refresh

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。



# 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

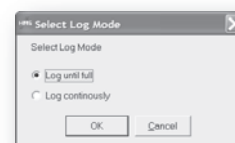
ご希望の操作のモードを選択し、'OK' をクリックしデータのログを開始します。

- 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。

Line #	Relative Time(s)	Hex	Dec	ASCII	Hex	Dec	ASCII
1	0				0x0A	10	
2	0				0x00	0	
3	0				0x00	0	
4	0				0x00	0	
5	1				0x00	0	
6	0				0x01	1	
7	0				0x05	5	
8	1				0x71	113	
9	4	0x0A	10				
10	0	0x03	3				
11	0	0x02	2				
12	1	0x00	0				
13	1	0x00	0				
14	0	0x1D	29				
15	0	0x05	5				
16	0				0x0A	10	
17	0				0x10	16	
18	1				0x01	1	
19	1				0x00	0	
20	0				0x00	0	
21	1				0x01	1	
22	0				0x02	2	
23	1				0x00	0	

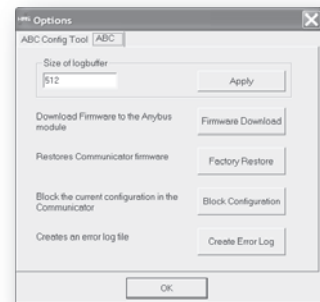
## コンフィグレーション

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い場合は、"Tools" 3-3 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-CompactCom Modbus-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**

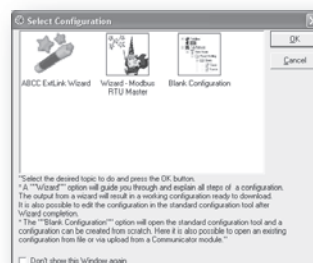
このオプションは Modbus RTU ベースのネットワークに最適です。

参照 ⇨ “ウィザード - Modbus RTU マスター”  
12-2

- **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。

**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があります。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。

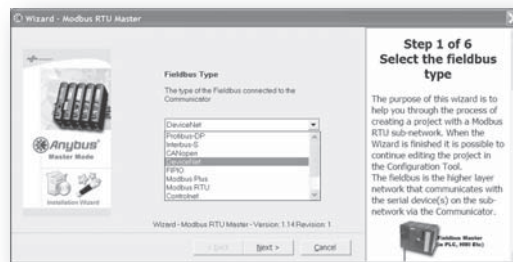
機器が要求する特定 Modbus コマンドをこのウィザードが処理しない場合、ABC コンフィグツールでこのコマンドをトランザクションとしてマニュアルで特定することができます。

ウィザード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

'DeviceNet' を選択し、'Next' をクリックします。

**ポイント** 'Previous' をクリックすると、現在の設定を維持したまま前のメニューに戻れます。

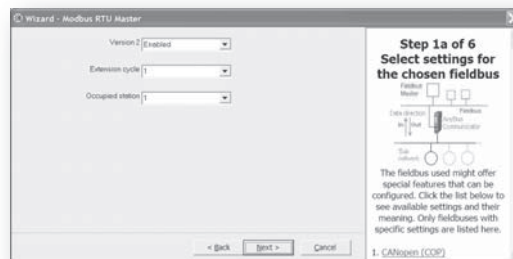


### ステップ 2 I/O サイズ

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。

'Next' をクリックします。

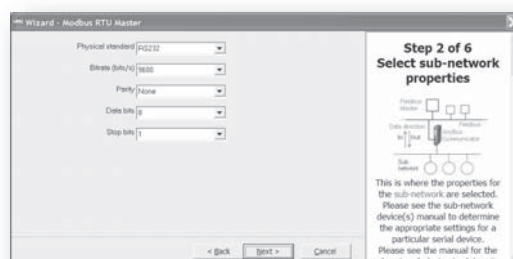
**参照** ⇨ “IO Sizes” 4-1



### ステップ 3 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、'Next' をクリックします。

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 (“Status / Control Word” 4-2 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

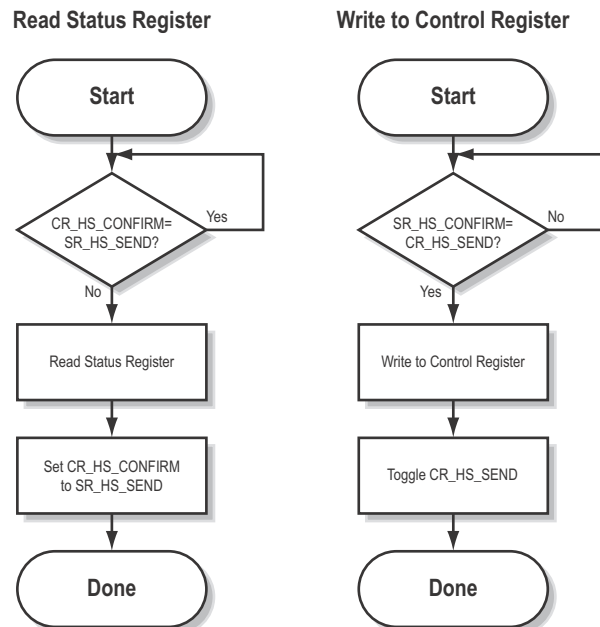
- フィールドバス制御システムへサブネットワークに関連する問題を報告する
- 双方向で有効なデータだけが交換されたことを確認する
- フィールドバス制御システムがサブネットワーク上の選択されたノードでデータ交換をスタート / 終了可能にする

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）

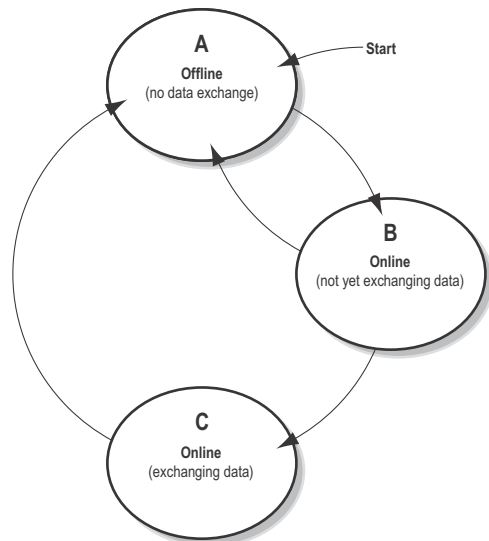
- 4.ゲートウェイがステータス・レジスタで'データ有効'ビットをクリアにするまで待つ
- 5.制御レジスタで'データ有効'ビットを設定
- 6.ゲートウェイがステータス・レジスタで'データ有効'ビットを設定すればステート C へ移行
- 7.フィールドバス上でゲートウェイがオフラインになるとステート A へ移行

### 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イクを制御します。
14	確認 (SR_HS_CONFIRM)	参照 ⇨ “ハンドシェイク手順” 13-2” “制御レジスタ構成（制御システムからゲートウェイへ）” 13-6
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  <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。
7... 0	データ	参照 ⇨ “マスター・モードでのステータス・コード” 13-5 “ジェネリック・データ・モードでのステータス・コード” 13-5

**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	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	I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・コンフィグ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。 <b>参照</b> ⇒ “コンスーム・トランザクション (Consume- Transactions)” 6-6
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'にならないかも知れません。

## 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)	これらのビットはゲートウェイのハンドシェークを制御します。 <b>参照</b> ⇨ “ハンドシェーク手順” 13-2 “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）” 13-4
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。 <b>参照</b> ⇨ 13-2" データの一貫性" 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。 <b>参照</b> ⇨ “マスター・モードでの制御コード” 13-6 “ジェネリック・データ・モードでの制御コード” 13-6
7... 0	Data	

**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、コンフィグ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

DeviceNet 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。

以下の 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" 14-2	14-2
" メッセージ・ルータ、クラス 02h" 14-4	14-4
"DeviceNet オブジェクト、クラス 03h" 14-5	14-5
" アセンブリ・オブジェクト、クラス 04h" 14-6	14-6
" 接続オブジェクト、クラス 05h" 14-7	14-7
" 承認ハンドラ・オブジェクト、クラス 2Bh" 14-10	14-10

### 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" 14-11	14-11
" 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" 14-12	14-12
" 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" 14-13	14-13

## 識別オブジェクト、クラス 01h

### 概論

#### オブジェクト説明

—

#### 実装されたサービス

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：003Ch	Anybus Communicator for DeviceNet
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	Module serial number	モジュールのシリアル番号
7	Get	Product Name	SHORT_STRING	“AnyBus-C”	製品名
8	Get	Config Consist Value	UINT	N/A	-

**Device Status**

ビット	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... 7	(リザーブド)
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: Get Attribute Single

インスタンス・サービス: —

### 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

—

## DeviceNet オブジェクト、クラス 03h

### 概論

#### オブジェクト説明

—

#### 実装されたサービス

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	0002h	Revision 2

### インスタンス属性

#	アクセス	名前	タイプ	値	説明
1	Get	MAC ID	USINT	N/A	現在使用されている MAC ID
2	Get	Baud Rate	USINT	N/A	現在使用されているボーレート 1=125 kbps 2=250 kbps 3=500 kbps
5	Get	Allocation Information	Struct of:		-
			BYTE	N/A	割り付け選択バイト
			USINT	N/A	マスターの MAC ID



## 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。

参照 ⇨ “DeviceNet オブジェクト、クラス 03h” 14-5  
 “フィールドバス設定” 4-1

#### 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	0001h	Revision 1

### 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。

#	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 – インスタンス / 接続ポイント 96h

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

#	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

## 接続オブジェクト、クラス 05h

### 概論

#### オブジェクト説明

—

#### 実装されたサービス

##### 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

インスタンス 1 = エクスプリシット・メッセージング接続  
(DeviceNet オブジェクト内で事前定義)

インスタンス 2 = ポールド接続 /COS/ サイクリック・コンスーミング接続

インスタンス 3 = ビット・ストローブ接続

インスタンス 4 = COS/ サイクリック・プロデュース接続

インスタンス 10 - 14 = エクスプリシット接続 (割り付けられた UCMM)

### インスタンス 1 & 10...14 (エクスプリシット・メッセージング接続) 属性

#	アクセス	名前	タイプ	値	説明
1	Get	State	USINT	N/A	0 = 現存しない 1 = コンフィグレーション中 3 = 完了 4 = タイムアウト 5 = 延期された削除
2	Get	Instance type	USINT	0	エクスプリシット・メッセージング接続

## インスタンス 2 (ポーリングされた接続) 属性

#	アクセス	名前	タイプ	値	説明
1	Get	State	USINT	N/A	0 = 現存しない 1 = コンフィグレーション中 3 = 完了 4 = タイムアウト
2	Get	Instance type	USINT	1	I/O 接続

## インスタンス 3 (ビット・ストローブ接続) 属性

#	アクセス	名前	タイプ	値	説明
1	Get	State	USINT	N/A	0 = 現存しない 1 = コンフィグレーション中 3 = 完了 4 = タイムアウト
2	Get	Instance type	USINT	1	I/O 接続

## インスタンス 4 (COS/ サイクリック接続) 属性

#	アクセス	名前	タイプ	値	説明
1	Get	State	USINT	N/A	0 = 現存しない 1 = コンフィグレーション中 3 = 完了 4 = タイムアウト
2	Get	Instance type	USINT	1	I/O 接続
3	Get	Transport Class trigger	BYTE	N/A	接続の動作を定義
4	Get	Produced Connection ID	UINT	N/A	伝送のための CAN ID
5	Get	Consumed Connection ID	UINT	N/A	受信のための CAN ID
6	Get	Initial Comm Characteristics	BYTE	0Fh (No ACK)	メッセージ・グループ 1 でのプロデュース、コンスームなし
				01h (ACK)	メッセージ・グループ 1 でのプロデュース、メッセージ・グループ 2 でのコンスーム
7	Get	Produced Connection Size	UINT	N/A	この接続を介して伝送されたバイト数
8	Get	Consumed Connection Size	UINT	0	この接続を介して受信されたバイト数
9	Get/Set	Expected Packet Rate	UINT	0	この接続に関連するタイミング

#	アクセス	名前	タイプ	値	説明
12	Get	Watchdog timeout action	USINT	N/A	0 = タイムアウトしたステートの変化 1 = 自動削除 2 = 自動リセット 3 = 延期された削除
13	Get	Produced Connection path length	UINT	0006h	作成された接続パス属性内のバイト数
14	Get	Produced Connection Path	EPATH	20 04 24 66 30 03h	この接続上でデータを作成しているアプリケーション・オブジェクト
15	Get	Consumed Connection path length	UINT	0004h	消費された接続パス長属性内のバイト数
16	Get	Consumed Connection Path	EPATH	20 2B 24 01h	この接続オブジェクトにより消費されたデータを受信するアプリケーション・オブジェクトの指定

## 承認ハンドラ・オブジェクト、クラス 2Bh

## 概論

オブジェクト説明

—

実装されたサービス

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	0001h	Revision 1
1	Get	Max Instance	UINT	0001h	最大インスタンス数

## インスタンス属性

#	アクセス	名前	タイプ	値	説明
1	Get/Set	Acknowledge Timer	UINT	16	再送するまでの承認待ち時間 (ms 単位)
2	Get/Set	Retry Limit	USINT	1	リトライ数の限界までの ACK タイムアウト数
3	Get/Set	Producing connection Instance	UINT	4	ACK ハンドラ・イベントにより通知される I/O アプリケーション・オブジェクトを作成するパスを含む接続インスタンス
4	Get	Ack List Size	Byte	-	Ack Path List を使用したデータのメンバーの最大数 0 = 動的
5	Get	Ack List	Array of USINT	N/A	Ack を受信するアクティブ接続インスタンスのリスト
6	Get	Data with Ack Path List Size	Byte	-	Ack Path List を使用したデータのメンバーの最大数 0 = 動的
7	Get	Data with Ack Path List	Array of USINT	N/A	接続インスタンス / コンシューミング・アプリケーション・オブジェクトの対のリスト

**注意！** インスタンス 1 は ACK された COS/ サイクリック接続を使用した時に作成されます。

## 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。

#### 実装されたサービス

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

#	アクセス	名前	タイプ	値
01h	Get	Module serial number	UDINT	シリアル番号
04h	Get	Module Software version	UINT	DeviceNet インターフェース・ソフトウェア・バージョン
0Fh	Get	Input I/O Size <sup>a</sup>	UINT	IO サイズ In
11h	Get	Input Total Size	UINT	入力データ領域合計サイズ (I/O + パラメータ・データ)
12h	Get	Output I/O Size <sup>a</sup>	UINT	IO サイズ Out
14h	Get	Output Total Size	UINT	出力データ領域合計サイズ (I/O + パラメータ・データ)

a. 参照 ⇨ “IO Sizes” 4-1

## 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。

参照 ⇒ “DeviceNet 上のデータ表示” 2-8

“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” 14-13

“フィールドバス設定” 4-1

“パラメータ・データの初期化（エクスプリシット・データ）” A-1

#### サポートされたサービス

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。

参照 ⇒ A-2 “DeviceNet へのマッピング入力パラメータ・データ”

#	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。

参照 ⇒ “DeviceNet 上のデータ表示” 2-8

“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” 14-13

“フィールドバス設定” 4-1

“パラメータ・データの初期化（エクスプリシット・データ）” A-1

#### サポートされたサービス

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。

参照 ⇒ “DeviceNet への出力パラメータ・データのマッピング” A-5

#	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
...	...	...	...	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



# 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。

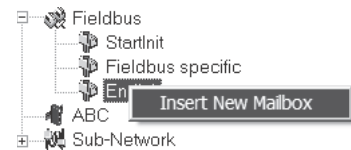
標準的な初期化パラメータはサブネットワーク・コンフィグレーションによって決定されます。サブネットワーク通信のための入力と出力データの合計についての情報は ABC コンフィグツールで使用されます。この情報によって ABC コンフィグツールは、ゲートウェイに組み込まれている Anybus-S インターフェースのデュアルポート RAM の入力と出力データ領域の大きさを設定するコンフィグレーション・メッセージを作成することができます。

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。

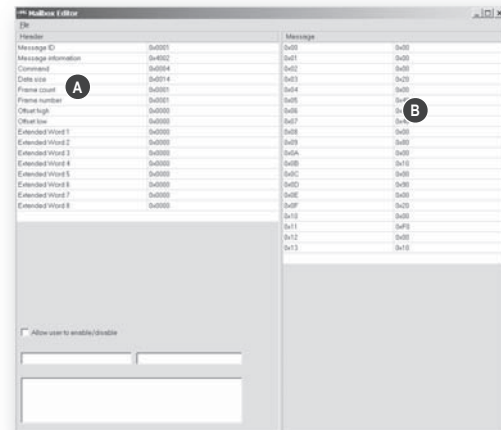
(メールボックス・メッセージは Anybus-S インターフェースを使用したローレベル通信向けに使用される HMS 特定コマンド・ストラクチャです。 Anybus-S Parallel Design Guide と使用するフィールドバスのフィールドバス・アペンディックスを参照してください。)

## 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、'Endlnit'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)

### 概要

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：

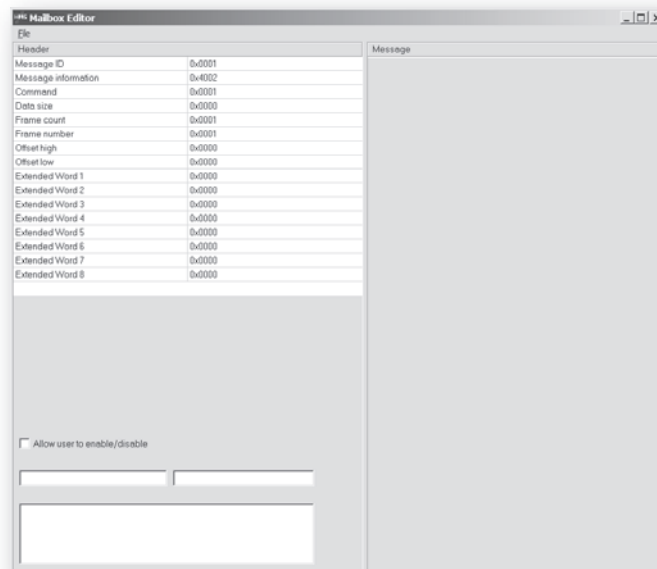
参照 ⇨ “先進フィールドバス・コンフィグレーション” 15-1

### 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。

参照 ⇨

“メールボックス・エディタ” 15-21

## DeviceNet 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。

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。  
参照 ⇒ “メールボックス・メッセージの追加” A-1
2. メールボックス・ヘッダ内の 'Command'- 値を 0004h 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 1 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アドレス・データを提供します。

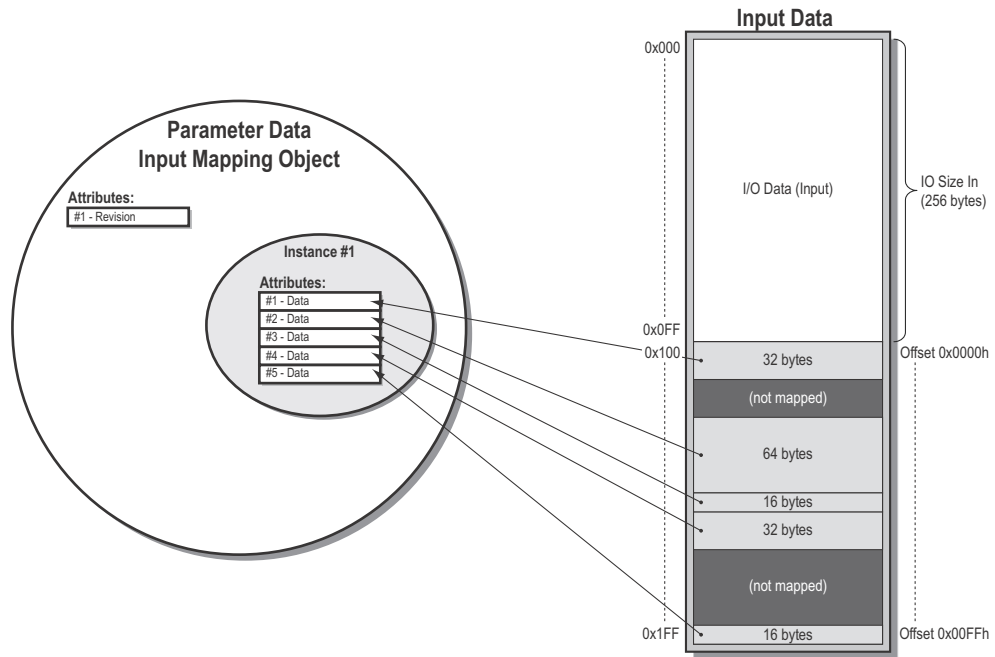
メールアドレス・データ		属性番号	コメント
割り付け	データ		
0x00	0x00	1	Offset = 0000h
0x01	0x00		
0x02	0x00		サイズ = 32 バイト
0x03	0x20		
0x04	0x00	2	Offset = 0040h
0x05	0x40		
0x06	0x00		サイズ = 64 バイト
0x07	0x40		
0x08	0x00	3	Offset = 0080h
0x09	0x80		
0x0A	0x00		サイズ = 16 バイト
0x0B	0x10		
0x0C	0x00	4	Offset = 0090h
0x0D	0x90		
0x0E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

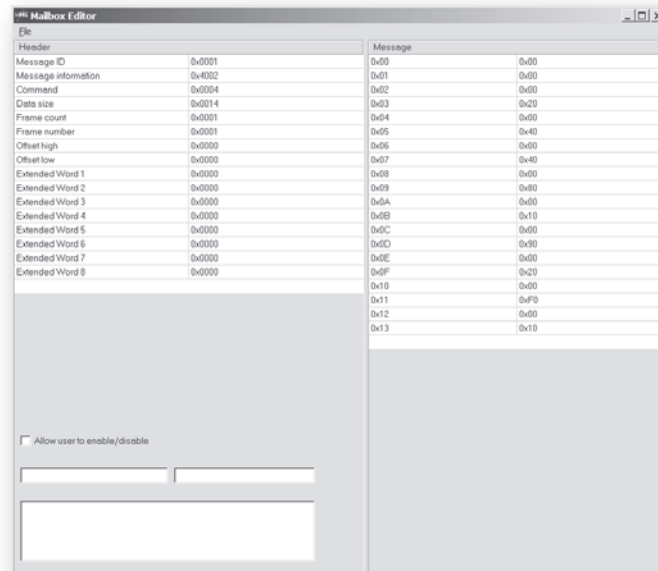
5. 新しいメールアドレスを保存するために 'File' メニューの 'Apply changes' を選択します。

※ 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



## DeviceNet 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。

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。

参照 ⇨ “メールボックス・メッセージの追加” A-1

2. メールボックス・ヘッダ内の 'Command'- 値を 0005h 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 1 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。これらの値はビック・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。

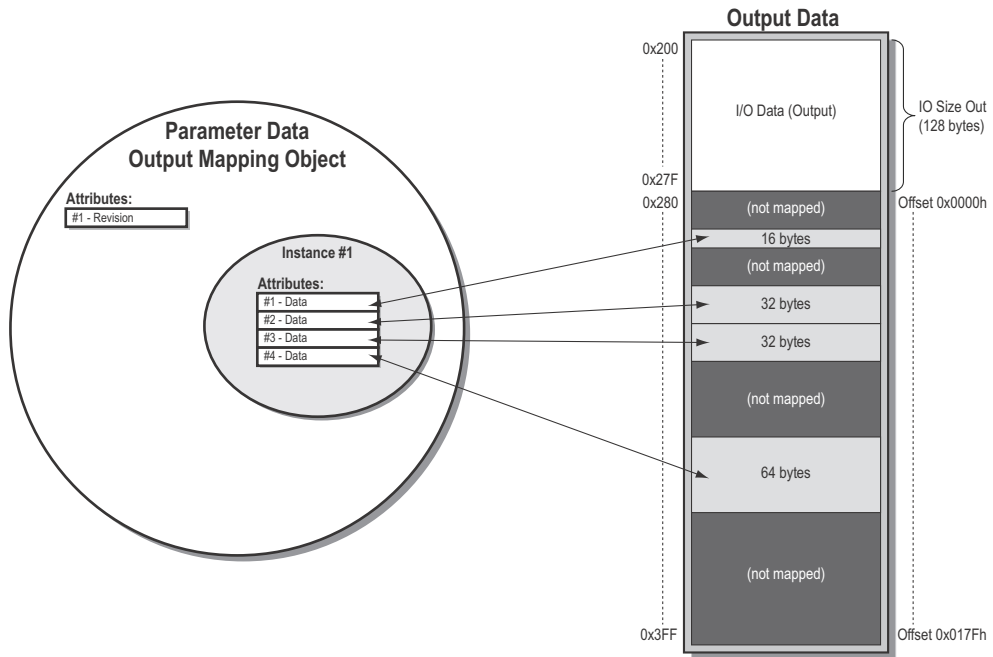
メールボックス・データ		属性番号	コメント
割り付け	データ		
0x00	0x00	1	Offset = 0020h  サイズ = 16 バイト
0x01	0x20		
0x02	0x00		
0x03	0x10		
0x04	0x00	2	Offset = 0050h  サイズ = 32 バイト
0x05	0x50		
0x06	0x00		
0x07	0x20		
0x08	0x00	3	Offset = 0070h  サイズ = 32 バイト
0x09	0x70		
0x0A	0x00		
0x0B	0x20		
0x0C	0x00	4	Offset = 00D0h  サイズ = 64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

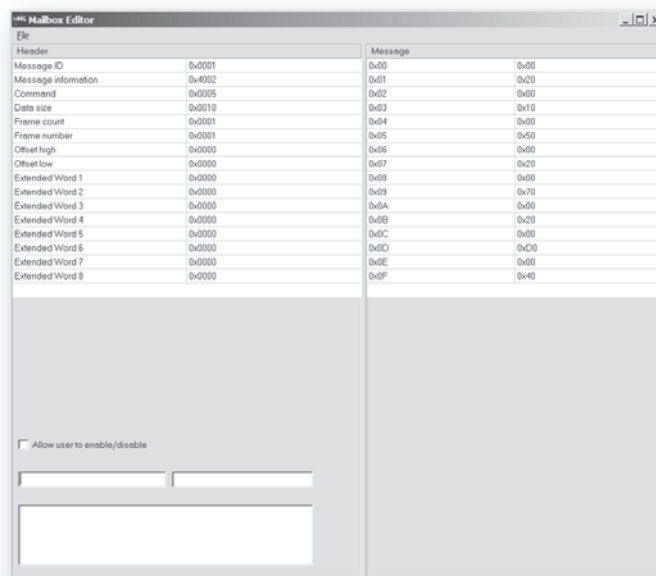
- 新しいメールボックスを保存するために 'File' メニューの 'Apply changes' を選択します。

※ 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

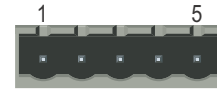




# コネクタ ピン割り付け

## DeviceNet Connector コネクタ

ピン	信号	説明
1	V-	DeviceNet bus 電源, ネガティブ電源圧
2	CAN L	Can L bus line
3	シールド	ケーブルシールド
4	CAN H	CAN H bus 線
5	V+	DeviceNet bus 電源, ポジティブ電源圧



## 電源コネクタ

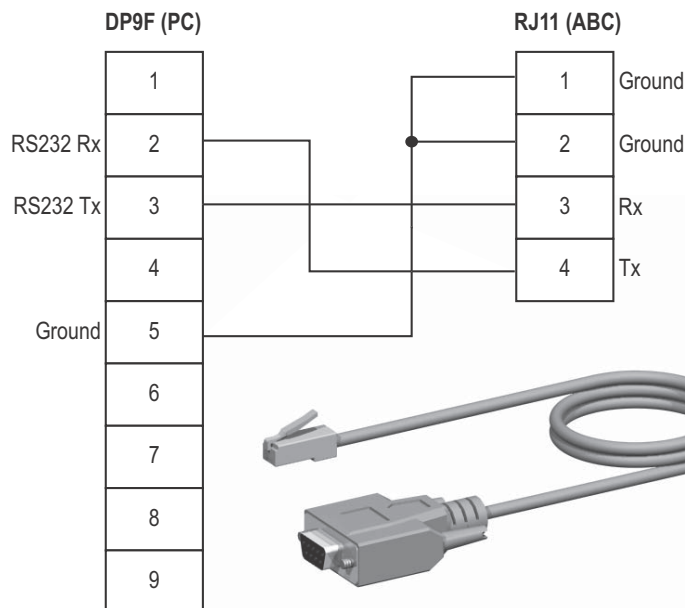
ピン	説明
1	+24V DC
2	GND



- 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。

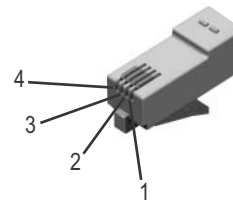
## PC コネクタ

コンフィグレーション・ケーブル・ワイヤリング

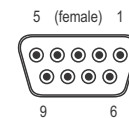


**RJ9 (ABC)**

ピン	説明
1	シグナル・クラウンド
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・クラウンド
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  $>200\text{mV}$  である閾値よりも高くなるように調節します。

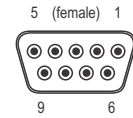
バイアス・レジスタは 1 つのノードにのみ実装されることに注意してください。いくつものノードにバイアス・レジスタを実装するとネットワーク上のシグナルの質について妥協してしまい、伝送問題を引き起こすこともあります。

**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  $100\text{...}120\text{R}$  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。

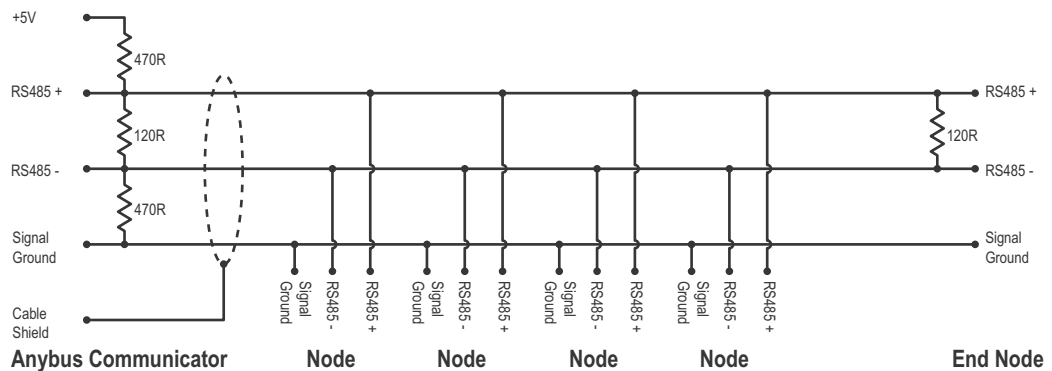
## コネクタ ピン配置 (DB9F)

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・クラウド <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	○	○	○

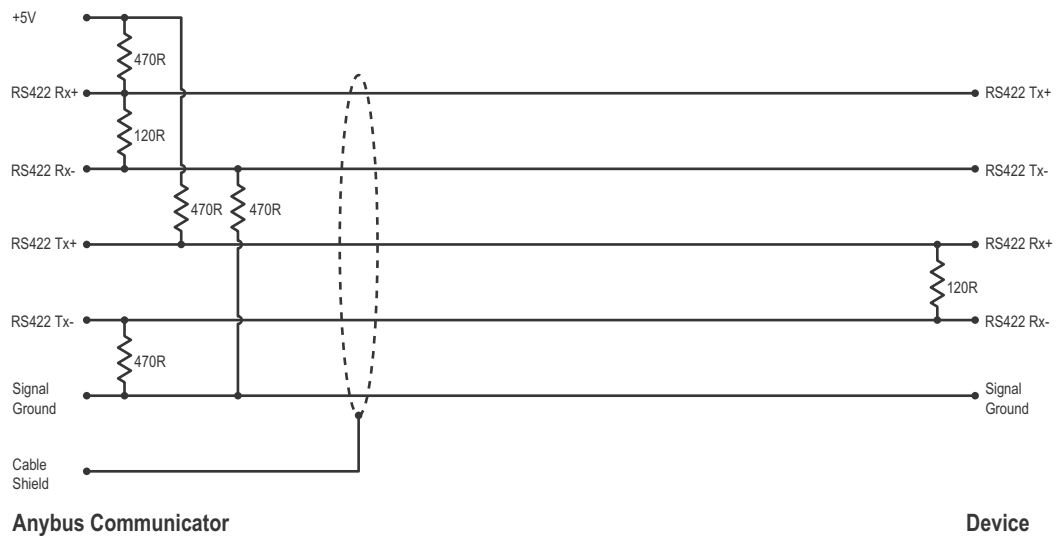


- a. グラウンド・ループなど、このシグナルを直接他のノードのプロテクトティブ・アースに接続する場合、オンボード・シリアル・トランシーバにダメージを与える場合があります。一般的には他のノードのシグナル・グラウンド (利用可能であれば) への接続のみを推奨します。

## 代表的な接続 (RS485)

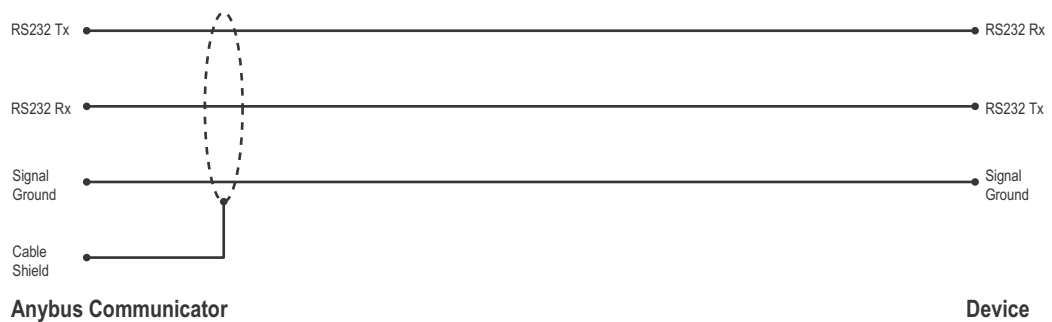


## 代表的な接続 (RS422 & 4-ワイヤ RS485)



**注意！** バイアス・レジスタは通常 RS422 では必要とされませんが、4-wire RS485 をお使になる場合は、必要とされる可能性があります。

## 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:  $\pm 0^{\circ}$  C から  $+55^{\circ}$  C

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

## 法的規制への遵守

### EMC 指令 (CE)

本製品は下記コンフォーマンスにより 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 IIIb

250 VRMS 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ト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	SO H 1	STX 2	ETX 3	EOT 4	EN Q 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	NA K 21	SYN 22	ETB 23	CA N 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



