



Anybus CompactCom 40

EtherNet/IP

NETWORK GUIDE

SCM-1202-031 2.1 ja-JP 日本語

必ずお読みください

免責条項

本ドキュメントの記載内容は、情報の提供のみを目的としています。記載内容に誤りや脱字があった場合は、HMS Networksにお知らせください。HMS Networksは、本ドキュメント内の誤りについて一切の責任を負いません。

HMS Networksは、製品開発に継続的に取り組むという自社のポリシーに基づき、製品に変更を加える権利を留保します。従って、本ドキュメントの記載内容はHMS Networksが保証するものではなく、予告なく変更される場合があります。HMS Networksは、本ドキュメントの記載内容を更新したり最新の状態に維持したりする義務を負いません。

本ドキュメントに提示されているデータ、例、図表は説明用のものであり、製品の機能と取り扱いについての理解を深めることのみを目的としています。本製品の使用にあたっては広範な用途が考えられ、個々の実装に関連して様々なバリエーションや要件が存在することから、本ドキュメントに提示されているデータ、例、図表に基づいた本製品の実際の使用について、また本製品の設置時に発生した損害について、HMS Networksは一切の責任を負いません。本製品の使用責任者は、本製品が当該のアプリケーションで正しく使用されること、ならびに適用されている法規、規制、規約、規格などを含めすべての性能要件および安全要件をそのアプリケーションが満たしていることを確保するために、十分な知識を取得する必要があります。さらに、HMS Networksは、いかなる場合であっても、本製品のドキュメントに記載されていない機能の使用や機能上の副作用によって生じた不具合について一切責任を負いません。本製品のそのような特徴を直接または間接に使用したことで生じる影響 (互換性の問題や安定性の問題など) は、本ドキュメントでは定義されていません。

目次

ページ

1	まえがき	5
1.1	本ドキュメントについて.....	5
1.2	関連ドキュメント.....	5
1.3	ドキュメント更新履歴.....	5
1.4	表記規則.....	6
1.5	ドキュメント固有の表記規則.....	6
1.6	略語.....	6
1.7	商標.....	7
2	Anybus CompactCom 40 EtherNet/IPについて	8
2.1	概要.....	8
2.2	特長.....	9
3	基本動作	10
3.1	基本情報.....	10
3.2	ネットワークID.....	11
3.3	通信設定.....	12
3.4	ビーコンベースDLR (デバイスレベルリング).....	14
3.5	ネットワークのデータ交換.....	14
3.6	Webインターフェース.....	15
3.7	電子メールクライアント.....	15
3.8	モジュラーデバイス機能.....	15
3.9	ファイルシステム.....	16
4	EtherNet/IP実装詳細	18
4.1	基本情報.....	18
4.2	EtherNet/IP & CIPの実装.....	18
4.3	アセンブリマッピングオブジェクト (EBh) の使用.....	19
4.4	ソケットインターフェース (上級者対象).....	21
4.5	診断.....	22
4.6	QuickConnect.....	22
4.7	CIPセーフティ.....	22
4.8	CIP Sync.....	23
5	FTPサーバ	24
5.1	基本情報.....	24
5.2	ユーザーアカウント.....	25
5.3	セッション例.....	26

6	Webサーバ	27
6.1	基本情報	27
6.2	デフォルト Webページ.....	27
6.3	サーバコンフィグレーション	32
7	電子メールクライアント	35
7.1	基本情報	35
7.2	電子メールメッセージを送る方法.....	35
8	Server Side Include (SSI)	36
8.1	基本情報	36
8.2	Include File	36
8.3	コマンドの機能	36
8.4	引数関数	51
8.5	SSI出力の設定	55
9	JSON	56
9.1	基本情報	56
9.2	JSONオブジェクト	57
9.3	例	75
10	CIPオブジェクト	76
10.1	基本情報	76
10.2	ステータスコードの変換.....	77
10.3	Identity オブジェクト (01h)	78
10.4	メッセージルータ (02h)	81
10.5	アセンブリオブジェクト (04h)	82
10.6	コネクションマネージャ (06h)	85
10.7	パラメータオブジェクト (0Fh)	89
10.8	Time Syncオブジェクト (43h)	92
10.9	DLRオブジェクト (47h)	96
10.10	QoS オブジェクト (48h)	97
10.11	Base Energy オブジェクト (4Eh)	98
10.12	Power Management オブジェクト (53h)	100
10.13	ADIオブジェクト (A2h).....	102
10.14	ポートオブジェクト(F4h).....	104
10.15	TCP/IPインターフェースオブジェクト (F5h)	106
10.16	Ethernet リンクオブジェクト (F6h)	109

11	Anybus モジュールオブジェクト	114
11.1	基本情報	114
11.2	Anybusオブジェクト (01h).....	115
11.3	診断オブジェクト (02h).....	116
11.4	Network オブジェクト (03h).....	117
11.5	ネットワークコンフィグレーションオブジェクト (04h)	118
11.6	ソケットインターフェースオブジェクト (07h).....	127
11.7	SMTPクライアントオブジェクト (09h).....	144
11.8	Anybusファイルシステムインターフェースオブジェクト (0Ah)	149
11.9	ネットワークイーサネットオブジェクト (0Ch).....	150
11.10	CIPポートコンフィグレーションオブジェクト (0Dh).....	152
11.11	ファンクショナルセーフティモジュール・オブジェクト (11h)	154
12	ホストアプリケーションオブジェクト.....	161
12.1	基本情報	161
12.2	Energy Reporting オブジェクト (E7h).....	162
12.3	ファンクショナルセーフティオブジェクト (E8h)	163
12.4	アプリケーション・ファイルシステムインターフェース・オブジェクト (EAh).....	165
12.5	CIP Identity ホストオブジェクト (EDh).....	166
12.6	SYNCオブジェクト (EEh).....	168
12.7	Energy Control オブジェクト (F0h).....	169
12.8	EtherNet/IPホストオブジェクト (F8h).....	175
12.9	Ethernet ホストオブジェクト (F9h)	184
A	機能の分類.....	189
A.1	基本	189
A.2	拡張	189
B	実装詳細.....	190
B.1	SUPビットの定義	190
B.2	Anybusステートマシン	190
B.3	Application Watchdog Timeout Handling.....	190
C	セキュアHICP (セキュアホストIPコンフィグレーション プロトコル).....	191
C.1	概要	191
C.2	操作	191

D	技術仕様.....	192
D.1	前面図.....	192
D.2	ファンクショナルアース (FE) の要件	193
D.3	電源	194
D.4	環境仕様	194
D.5	EMC準拠.....	194
E	タイミングと性能.....	195
E.1	基本情報	195
E.2	内部タイミング	195
F	コンFORMANCEテストガイド.....	197
F.1	概要	197
F.2	推奨のテストツール.....	197
F.3	適合宣言 (STC).....	200
G	後方互換性.....	215
G.1	初期段階における注意事項.....	215
G.2	ハードウェアの互換性	216
G.3	ソフトウェア全般	221
G.4	ネットワーク固有 – EtherNet/IP	223
H	ライセンス情報.....	225

1 まえがき

1.1 本ドキュメントについて

本ドキュメントは、Anybus CompactCom 40 EtherNet/IPの機能を十分理解していただくためのものです。本ドキュメントでは、Anybus CompactCom 40 EtherNet/IPの機能についてのみ説明しています。Anybus CompactComに関する一般的な情報については、Anybus CompactComのデザインガイドを参照してください。

本ドキュメントの読者は、ソフトウェア設計や通信システム全般に関して高い知識を備えていることが求められます。通常は、設計を実装するにはこのネットワークガイドの情報だけで十分です。ただし、EtherNet/IP固有の高度な機能を使用する場合は、EtherNet/IPネットワーキングの内部詳細、および公式のEtherNet/IP規格に関する十分な知識が必要となります。本製品の使用者は、EtherNet/IP規格を入手して十分な知識を得るか、または規格に関する知識を必要としない範囲で本製品を使用してください。

追加的文書やダウンロードファイルについてはwww.anybus.com/supportのサポートWebサイトをご覧ください。

1.2 関連ドキュメント

ドキュメント	作成者	ドキュメントID
Anybus CompactCom 40 Software Design Guide	HMS	HMSI-216-125
Anybus CompactCom M40 Hardware Design Guide	HMS	HMSI-216-126
Anybus CompactCom B40 Design Guide	HMS	HMSI-27-230
Anybus CompactCom Host Application Implementation Guide	HMS	HMSI-27-334
CIP仕様書、Volume 1 (CIP Common) と2 (EtherNet/IP)	ODVA	

1.3 ドキュメント更新履歴

1.1	2017/01/18	FMからDOXに、文書番号をHMSI-27-212からSCM-1202-031に変更。バージョン番号を初めからつけ直し。 M12コネクタを追加 小規模な修正と更新
1.2	2017/05/23	Ethernetホストオブジェクトに関する更新 (DHCPの無効化について) ポートオブジェクトに関する更新
1.3	2017/07/11	後方互換性に関するAppendix追加 TCP/IPインターフェースオブジェクト (CIP) に関する更新
1.4	2017/11/28	アセンブリマッピングオブジェクトガイド追加
1.5	2017/12/15	著作権付録を更新
1.6	2018/05/07	パフォーマンステストガイド追加 CIPオブジェクトADIおよびQoSに関する更新 各種更新
1.7	2018/09/19	EtherNet/IPホストオブジェクトのSet_Configuration_Dataに関する説明更新
1.8	2018/10/23	小規模な変更
1.9	2019/06/10	商標変更 CIP Sync追加 各種更新
2.0	2016/10/11	日本語に翻訳
2.1	2020/09/23	日本語に翻訳

1.4 表記規則

番号の付いたリストは、順番に実行する必要がある作業を示します。

1. まずこれを行います
2. その後これを行います

箇条書きのリストは以下を示します。

- 任意の順序で実行できる作業
- 項目化された情報
- ▶ このアクションは...
 - この結果につながります

User interaction elements (ボタンなど) は太字で表記されています。

```
Program code and script examples
```

本ドキュメント内の相互参照: [表記規則, ページ 6](#)

外部リンク (URL): www.hms-networks.com



警告

死亡または重篤な障害のリスクを避けるために従わなければならない指示。



注意

個人の負傷のリスクを避けるために従わなければならない指示。



機能の低減および/または機器への損傷のリスクを避けるため、またはネットワークのセキュリティのリスクを避けるために従わなければならない指示。



インストールおよび/または操作を容易にする可能性のある追加情報。

1.5 ドキュメント固有の表記規則

- 「Anybus」または「module」(モジュール)という表現はAnybus CompactComモジュールを表します。
- 「host」(ホスト)または「host application」(ホストアプリケーション)という表現はAnybus機器を制御する機器を表します。
- 16進数はNNNNhまたは0xNNNNの形式で表します。ここで、NNNNは16進の値を表します。
- 1バイトは常に8ビットで構成されます。
- "基本"と"拡張"を使用してオブジェクト、インスタンス、およびアトリビュートを分類します。

1.6 略語

略語	意味
API	assigned packet interval
RPI	requested packet interval
T	target (ここではモジュール)
O	origin (ここではマスター)

1.7 商標

Anybus®はHMS Industrial Networksの登録商標です。

EtherNet/IPはODVA, Inc.の商標です。

その他の商標は、各所有者に帰属します。

2 Anybus CompactCom 40 EtherNet/IPについて

2.1 概要

Anybus CompactCom 40 EtherNet/IP通信モジュールは、特許技術であるAnybus CompactComホストインターフェースを介して、EtherNet/IPへの適合性が確認された通信を簡単に実現します。この規格をサポートする装置はすべて、モジュールにより提供される機能を活用して、そのネットワークタイプを問わず、あらゆるネットワークにおいてシームレスなネットワーク統合を可能にします。モジュールは、リニアおよびリングネットワークトポロジ (DLR、デバイスレベルリング) の両方をサポートします。

Anybus CompactCom 40プラットフォームのモジュール形式アプローチは、特定のアプリケーション要件を満たすためにCIPオブジェクト実装を拡張することが可能となります。さらに、Identity オブジェクトがカスタマイズできるため、最終製品を汎用のAnybusモジュールではなく、ベンダー固有の実装のように見せることができます。

本製品は、Anybus CompactCom 40 Hardware / Software Design Guideに定義された、Anybus CompactCom 40モジュールのためのホストインターフェースにあらゆる点において対応しています。そのため、その規格に準拠したあらゆる機器と完全に置き換え可能です。通常、ネットワークに関連したソフトウェアを別途用意する必要はありませんが、高度なネットワーク固有の機能をすべて活用するためには、専用のソフトウェアが必要となる場合があります。

2.2 特長

- 2つのEtherNet/IPポート
- イーサネットコネクタまたはM12コネクタ
- 最大のリードプロセスデータ：1448バイト
- 最大のライトプロセスデータ：1448バイト
- 最大のプロセスデータ (リード+ライト、単位：バイト)：2896バイト
- ビーコンベースDLR (デバイスレベルリング)とリニアネットワークトポロジに対応
- ブラックチャネルインターフェース。これでCIP Safetyに対応する透過チャネルを提供します。
- 10/100 Mbit、全二重/半二重通信
- CIP Sync機能
- コンテンツのカスタム化が可能なWebサーバ
- FTPサーバ
- Eメールクライアント
- Server Side Include (SSI) 機能
- JSON機能
- カスタム化可能な識別情報
- 最大65535 ADI
- CIPパラメータオブジェクトに対応
- CIPオブジェクトの実装を拡張可能
- 未接続CIPルーティングに対応
- 透過ソケットインターフェース
- モジュラーデバイス機能
- QuickConnect対応
- 複数のIOアセンブリインスタンスの作成が可能

3 基本動作

3.1 基本情報

3.1.1 ソフトウェアの要件

Anybus CompactCom 40 EtherNet/IPをサポートするために、ネットワーク対応のコードを新たに記述する必要はありません。ただし、EtherNet/IPネットワークシステムの特性上、以下のようないくつかの制約を考慮する必要があります。

- モジュールにある一部の機能は、コマンド `Get_Instance_Number_By_Order` (アプリケーション データ オブジェクト, FEh)がホストアプリケーションに実装済みであることを前提要件としています。
- 通常の動作中にホストアプリケーションが作成できる診断インスタンスは最大5つ ([診断オブジェクト \(02h\)](#), [ページ 116](#)参照) です。重大な障害が発生した場合には、6つ目のインスタンスを追加で作成できます。この限界値はネットワークによってではなく、モジュールで設定されます。
- EtherNet/IP自体は、非周期的要求 (アプリケーションデータオブジェクトのインスタンス宛て要求など) については何らかの特定のタイミング要求を課すことはありませんが、原則的にこうした要求には適度な時間以内に処理して応答することが推奨されます。要求を送るアプリケーションはタイムアウトも決定します。例えば、EIPScanはタイムアウト10秒を使用しています。
- CIP固有の高度な機能を使用するには、CIPネットワークの内部詳細情報や公式のCIPおよびEtherNet/IP仕様から得られる情報が必要です。本製品の实装担当者はそうした場合にはこれらの規格を入手して十分な知識を得るか、規格に関する知識を必要としない範囲で実装を制限してください。

下記も参照してください。

- [診断オブジェクト \(02h\)](#), [ページ 116](#) (Anybusモジュールオブジェクト)
- 『Anybus CompactCom 40 Software Design Guide』、「アプリケーションデータオブジェクト(FEh)」

Anybus CompactComのソフトウェアインターフェースに関する詳細は、『Anybus CompactCom 40 Software Design Guide』を参照してください。

3.1.2 Electronic Data Sheet (EDS)

EtherNet/IPでは、デバイスの特性がASCIIデータファイル (拡張子EDS) として保存されています。このファイルは、ネットワークコンフィグレーションを設定する際にコンフィグレーションツールなどによって使用されます。HMS Industrial Networksは、モジュールのデフォルト設定に対応する標準 (ジェネリック) EDSファイルを提供します。しかし、Anybus CompactComコンセプトの本質は柔軟性に富むので、ジェネリックEDSファイルを無効にして製品の動作態様を変更することが可能です。このような場合にはカスタム EDS ファイルを作成する必要があります。これは、デフォルトの識別情報を無効にして、再認証が必要になります。

モジュールがパラメータオブジェクトを実装するので、RSNetWorxなどの設定ツールが適切なEDSファイルを自動的に生成することができます。この機能は、コマンド `Get_Instance_Number_By_Order` (アプリケーション データオブジェクト, FEh)がホストアプリケーションに実装済みであることが必要です。

下記も参照してください。

- [パラメータオブジェクト \(0Fh\)](#), [ページ 89](#) (CIPオブジェクト)
- 『Anybus CompactCom 40 Software Design Guide』、「アプリケーションデータオブジェクト(FEh)」



HMS Industrial Networksが標準EDSファイルの使用を許諾するのは、それが実際の実装に一致する場合、ならびに識別情報が不変のままである場合に限られます。

3.2 ネットワークID

デフォルトでは、モジュールは以下の識別設定を用います。

ベンダーID:	005Ah (HMS Industrial Networks)
デバイスタイプ:	002Bh (ジェネリックデバイス)
プロダクトコード:	0037h (Anybus CompactCom 40 EtherNet/IP)
プロダクト名:	「Anybus CompactCom 40 EtherNet/IP(TM)」

オプションで、EtherNet/IP ホストオブジェクトの中の対応するインスタンスアトリビュートを実装すればモジュールの識別情報をカスタム化することができます。

下記も参照してください。

- [Identity オブジェクト \(01h\), ページ 78](#) (CIPオブジェクト)
- [EtherNet/IPホストオブジェクト \(F8h\), ページ 175](#) (ホストアプリケーションオブジェクト)



CIP仕様に従い、ベンダーIDとシリアル番号の組合せは一意でなければなりません。カスタムシリアル番号をHMSベンダーID (005Ah) と組合せて使用することは許可されておらず、ベンダーIDを恣意的に選択することもできません。この要件に従わない場合、相互運用性上の問題やその他の望ましくない副作用が誘発されます。

ベンダーIDについてはODVAにお問い合わせください。

3.3 通信設定

ネットワーク関連の通信設定はネットワークコンフィグレーションオブジェクト (04h) の中にグループ化されており、次の内容を含みます。

IP設定	これらの設定は、モジュールがネットワークに参加できるよう正しく行わなければなりません。 モジュールはDHCPをサポートするため、IP設定はDHCPサーバから自動的に取得されます。DHCPはデフォルトでは有効ですが、必要に応じて無効にできます。
物理リンク設定	デフォルトではモジュールはオートネゴシエーションを使用して物理リンクを設定しますが、必要であれば特定の設定を強制することが可能です。

ネットワークコンフィグレーションオブジェクト(04h)のパラメータは、内蔵WebサーバおよびTCP/IPインターフェースオブジェクト (CIP) を通してネットワークから取得できます。

下記も参照してください。

- [Webサーバ, ページ 27](#)
- [TCP/IPインターフェースオブジェクト \(F5h\), ページ 106](#) (CIPオブジェクト)
- [Ethernet リンクオブジェクト \(F6h\), ページ 109](#) (CIPオブジェクト)
- [ネットワークコンフィグレーションオブジェクト \(04h\), ページ 118](#) (Anybusモジュールオブジェクト)
- [セキュアHICP \(セキュアホストIPコンフィグレーション プロトコル\), ページ 191](#)

3.3.1 スタンドアロンシフトレジスタ モードにおける通信設定

Anybus CompactComをスタンドアロンとして使用する場合は、IPアドレスを設定するためのアプリケーションはありません。IPアドレスは、この代わりに、DIP1 スイッチ(IPアドレス バイト 3)と、セットアップ中にメモリーに書き込まれる(IPアドレス バイト 0 - 2)仮想アトリビュート (Ethernetホストオブジェクト (F9h)、アトリビュート#17)を使用して設定されます。下記のフローチャートを参照してください。

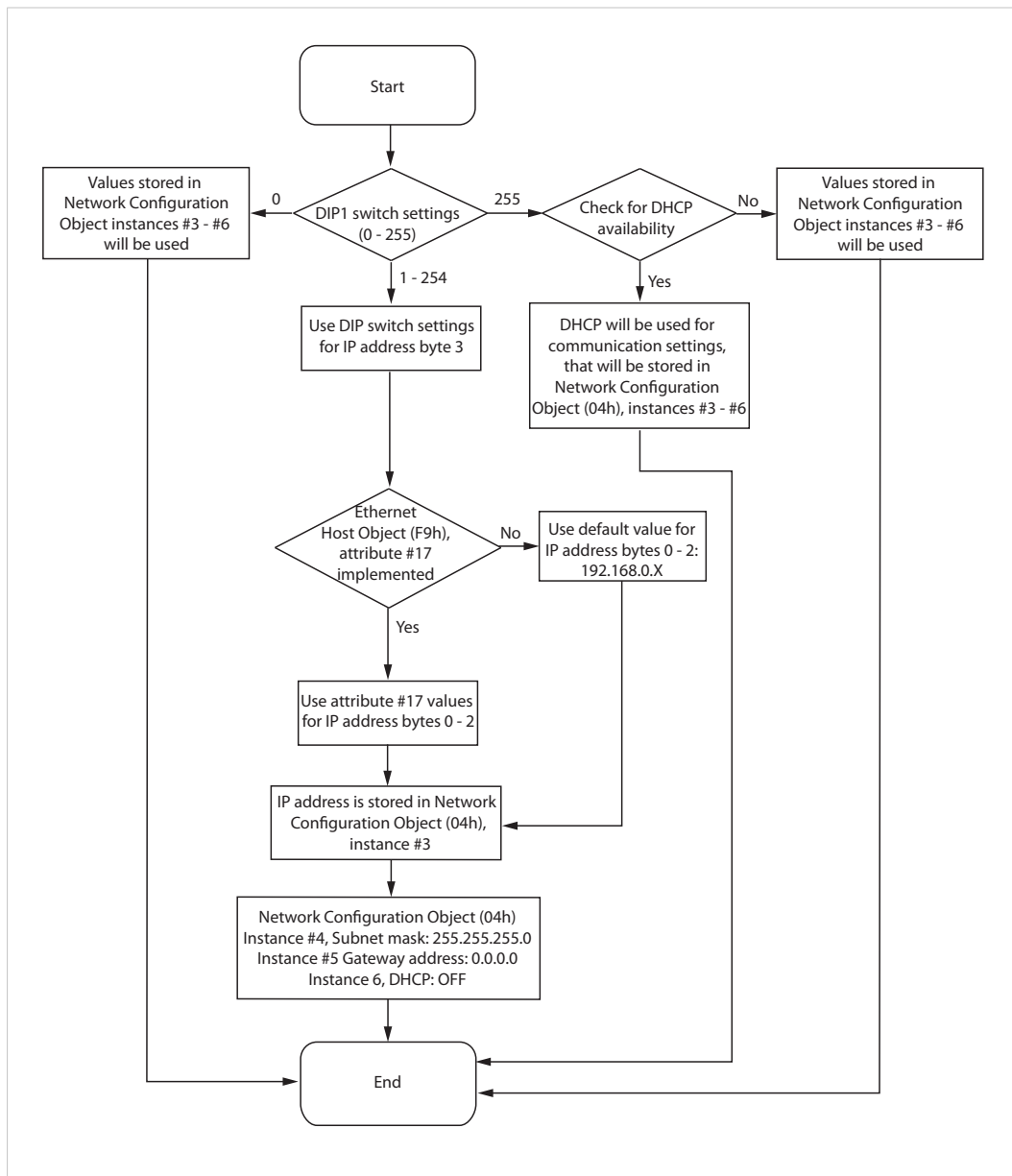


図 1

下記も参照してください。

- [Ethernet ホストオブジェクト \(F9h\), ページ 184](#)
- Anybus CompactCom M40 Hardware Design Guide
- [ネットワークコンフィグレーションオブジェクト \(04h\), ページ 118](#)

3.4 ビーコンベースDLR (デバイスレベルリング)

デバイスレベルリング (DLR)は、プログラム可能オートメーションコントローラや I/O モジュールなどオートメーション端末デバイスの組み込みスイッチ機能を使用する産業用途用のネットワーク技術であって、デバイスレベルでのEthernetリングネットワークポロジを可能にします。DLR技術はネットワークを強靱なものにし、マシンの最適な運用を可能にします。ビーコンベースのDLRネットワークはリングスーパーバイザと多数のリングノードで構成され、「ビーコン」を使用してリングの中断を検出します。DLRネットワークがリング内の中断を検出すると、データの代替経路を提供してネットワークを復旧します。DLR製品に内蔵された診断が故障箇所を特定できるので修復時間を削減するのに役立ちます。Anybus CompactCom 40 EtherNet/IPはDLRプロトコルを実装します。これはデフォルトで有効になっています。デバイスはリングスーパーバイザから送られてくるビーコンを処理してこれに作用することができ、100 μsまでのビーコンスピードに対応します。(必要な) DLR 機能は無効にできます。EtherNet/IPホストオブジェクトでアトリビュート#31 (Enable DLR) をFalse (無効) に設定することによってこれを無効にできます。 [EtherNet/IPホストオブジェクト \(F8h\)](#), ページ 175参照。

3.5 ネットワークのデータ交換

3.5.1 アプリケーション データ

アプリケーション データ インスタンス (ADI) はADIオブジェクト (CIP)によって表されます。このオブジェクトの中の各インスタンスはホストアプリケーション側のアプリケーション データオブジェクト内のインスタンスに直接対応しています。

ADIのアクセス可能な範囲は1 ~ 65535です。

下記も参照してください。

- [パラメータオブジェクト \(0Fh\)](#), ページ 89 (CIPオブジェクト)
- [ADIオブジェクト \(A2h\)](#), ページ 102 (CIPオブジェクト)

3.5.2 プロセスデータ

プロセスデータはアセンブリオブジェクト (CIP)の中で専用インスタンスとして表象されています。

下記も参照してください。

- [アセンブリオブジェクト \(04h\)](#), ページ 82 (CIPオブジェクト)
- [コネクションマネージャ \(06h\)](#), ページ 85 (CIPオブジェクト)

3.5.3 データ型の変換

Anybusデータ型は以下に示すようにCIP規格のデータ型に変換されます。逆の変換も行われます。

Anybusデータ型	CIPデータ型	コメント
BOOL	BOOL	この種の各ADI 要素は1バイトを占有します。
ENUM	USINT	
SINT8	SINT	
UINT8	USINT	
SINT16	INT	この種の各ADI 要素は2バイトを占有します。
UINT16	UINT	
SINT32	DINT	この種の各ADI 要素は4バイトを占有します。
UINT32	UDINT	
FLOAT	REAL	
CHAR	SHORT_STRING	SHORT_STRINGは、1バイト長のフィールド (この場合には ADI 要素数を表す)と、これに続く実際の文字データ (この場合は実際のADI 要素)で構成されます。これは10文字の文字列が11 バイトを占めることを意味します。
SINT64	LINT	この種の各ADI 要素は8バイトを占有します。
UINT64	ULINT	
BITS8	バイト	この種の各ADI 要素は1バイトを占有します。
BITS16	WORD	この種の各ADI 要素は2バイトを占有します。
BITS32	DWORD	この種の各ADI 要素は4バイトを占有します。
OCTET	USINT	

Anybusデータ型	CIPデータ型	コメント
BITS1-7	バイト	サイズ1~7のビットフィールド
PAD0-8	バイト	パディングで使用するサイズ0~8のビットフィールド
PAD9-16	バイト	パディングで使用するサイズ9~16のビットフィールド
BOOL1	BOOL	

3.6 Webインターフェース

Webインターフェースは特定のアプリケーションに合わせて完全にカスタム化可能です。ダイナミックなコンテンツをJSONとSSIスクリプトによって作成できます。データとWebページはFLASHをベースにしたファイルシステムに保存され、ここへは標準的なFTPクライアントを使用してアクセスできます。

下記も参照してください。

- [ファイルシステム, ページ 16](#)
- [FTPサーバ, ページ 24](#)
- [Webサーバ, ページ 27](#)
- [Server Side Include \(SSI\), ページ 36](#)
- [JSON, ページ 56](#)

3.7 電子メールクライアント

内蔵電子メールクライアントは、ホストアプリケーションがファイルシステムに保存されているメールメッセージを送信することを可能にします。または、SMTPクライアントオブジェクト (09h)の中に直接定義できます。メッセージはSSIコンテンツに関してスキャンされます。即ち、動的情報をファイルシステムから埋め込むことができます。

下記も参照してください。

- [ファイルシステム, ページ 16](#)

3.8 モジュラーデバイス機能

モジュラーデバイスは、ある一定数のスロットを持つバックプレーンで構成されます。先頭のスロットは、Anybus CompactComモジュールを含む「カバー」で占められています。その他のすべてのスロットは、空であるかモジュールで占められています。

ADIをプロセスデータにマッピングするとき、アプリケーションは、各モジュールのプロセスデータをスロット順にマッピングします。

モジュラーデバイス内のモジュールリストは、CIP Identityオブジェクトへの要求によって、EtherNet/IPネットワークマスターに対して利用可能になります。

下記も参照してください。

- “モジュラーデバイスオブジェクト(ECh)” (『Anybus CompactCom 40 Software Design Guide』を参照)
- [Identity オブジェクト \(01h\), ページ 78](#) (CIPオブジェクト)

3.9 ファイルシステム

3.9.1 要約

Anybus CompactCom 40 EtherNet/IPにはアプリケーションからもネットワークからもアクセス可能なファイルシステムを搭載しています。以下の3つのディレクトリがあらかじめ定義されています。

- VFS** 仮想ファイルシステム。モジュールのWebページを保持するなどします。仮想ファイルシステムは、デフォルトでAnybusファイルシステムインターフェースオブジェクト(OAh)により有効になっています。
- Application** アプリケーション・ファイルシステムインターフェース・オブジェクト (EAh) (オプション) を介して、このディレクトリにあるアプリケーションのファイルシステムにアクセスできます。
- Firmware** ファームウェアの更新ファイルはこのディレクトリに保存されます。

i ファームウェアフォルダでは、ファイルの書き込み中に挿入モードは使用できません。必ず書き込みモードだけ使用してください。

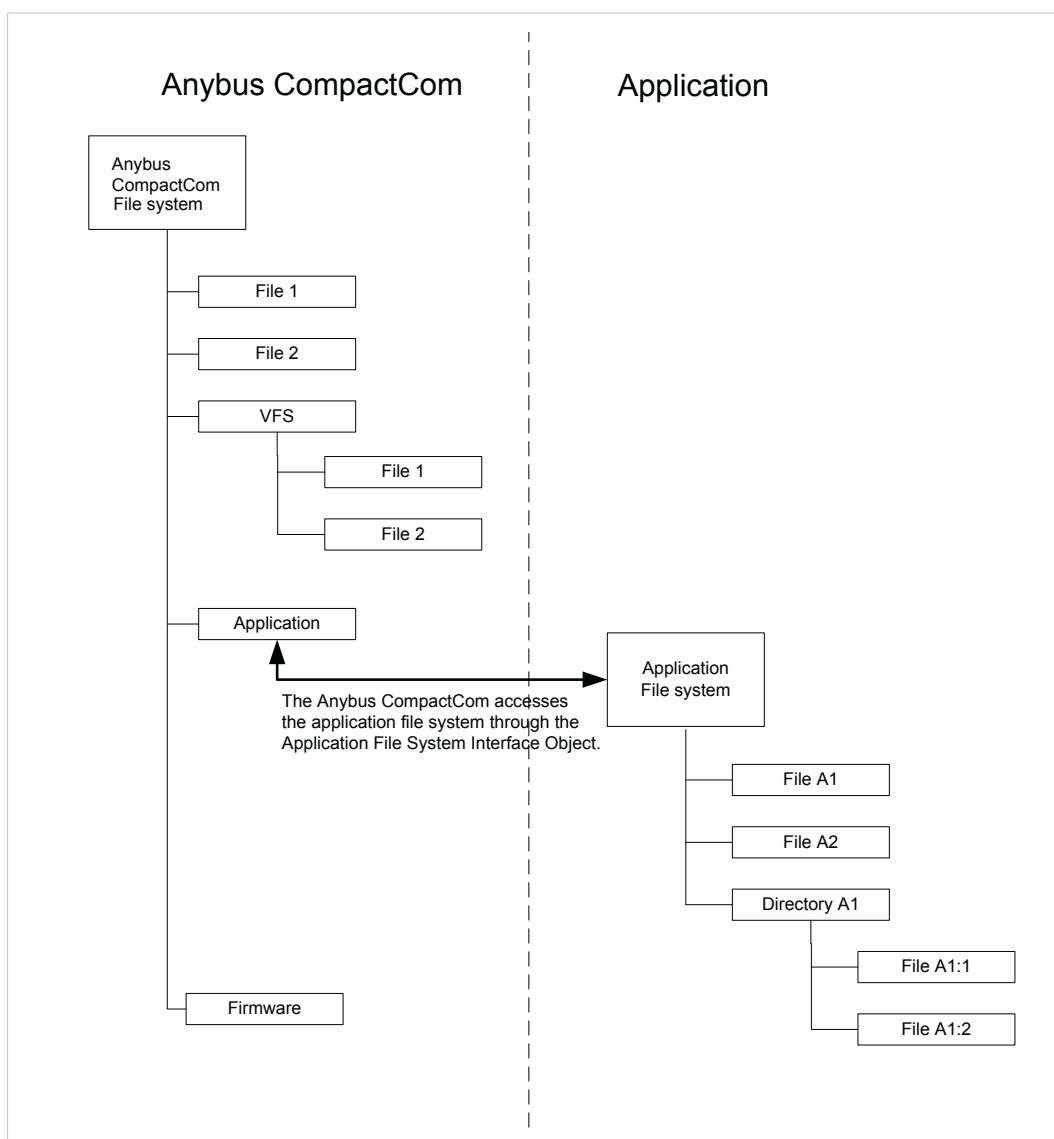


図 2

3.9.2 基本情報

内蔵ファイルシステムは28 Mバイトの不揮発ストレージのホストとなっており、ここへはHTTPやFTPサーバ、Eメールクライアント、(Anybusファイルシステムインターフェースオブジェクト(0Ah)を介して) ホストアプリケーションからアクセスできます。

ルートディレクトリに保存できるディレクトリとファイルの最大数は、ファイル名が短い場合 (ファイル名8 バイト + 拡張子3 バイト) に限り511です。ルート以外のディレクトリには無数のファイルを保存できます。

ファイルシステムは以下の命名規則に従います:

- \ (バックスラッシュ) はパスの分離子として使用します。
- 名前には空白を含むことができますが、空白は名前の最初と最後には使用できません。
- 名前に使える文字は127文字のASCII文字と数字であって、次の文字は使えません: \ / : * ? " < > |
- 名前の長さは48文字を超えることはできません
- パスは126文字より長いことはできません (ファイル名を含む)

下記も参照してください。

- [FTPサーバ, ページ 24](#)
- [Webサーバ, ページ 27](#)
- [電子メールクライアント, ページ 35](#)
- [Server Side Include \(SSI\), ページ 36](#)
- [Anybusファイルシステムインターフェースオブジェクト \(0Ah\), ページ 149](#)
- [アプリケーション・ファイルシステムインターフェース・オブジェクト \(EAh\), ページ 165](#)



ファイルシステムはフラッシュメモリにあります。技術的な理由により各フラッシュセグメントは約100000回消去するとエラーが出始め、*random access storage*の用途に適さなくなります。

以下の処理は1つ以上のフラッシュ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削除、移動、名前変更
- 既存ファイルへのデータ上書や挿入
- ファイルシステムのフォーマット

3.9.3 システムファイル

ファイルシステムはシステム設定用のファイル一式を含みます。これらのファイルは「システムファイル」と呼ばれ、標準のテキストエディタ (Microsoft Windows™のノートパッドなど) で編集できる通常のASCIIファイルです。いくつかの例外を除き、これらのファイルの形式はKeyの概念をベースにしており、各Keyに1つの値を代入することができます。下記を参照してください。

例 1:

```
[Key1]
value of Key1

[Key2]
value of Key2
```

4 EtherNet/IP実装詳細

4.1 基本情報

この章では、Anybus実装におけるEtherNet/IPの個別詳細を説明します。これらの機能を使用するには、EtherNet/IPネットワークの内部詳細や、公式のEtherNet/IPおよびCIP規格に関する十分な知識が必要です。こうした場合には本製品の实装担当者は、同規格を入手して十分な知識を得るか、または規格に関する知識を必要としない範囲に実装を限ってください。EDS ファイルはすべての変更を反映するように変更する必要があります。

4.2 EtherNet/IP & CIPの実装

デフォルトではこのモジュールはジェネリックなCIP プロファイルをサポートします。オプションとして、未実装のCIPオブジェクトへの要求をホストアプリケーションヘルト変更する事が可能です。それにより、別プロファイルに対応すること等が可能です。

特定のプロファイルに対応させるには以下の手順を行います：

1. プロファイルの要件に従ってアイデンティティ設定をEtherNet/IP ホストオブジェクトにセットアップします。
2. アセンブリマッピングオブジェクトをホストアプリケーションに実装します。
3. アセンブリ インスタンスの数を プロファイルの要件に従ってセットアップします。
4. EtherNet/IP ホストオブジェクトにあるホストアプリケーションへのCIPメッセージのルーティングを有効にします。
5. 必要なCIPオブジェクトをホストアプリケーションに実装します。

下記も参照してください。

- [アセンブリマッピングオブジェクト \(EBh\) の使用, ページ 19](#)
- [EtherNet/IPホストオブジェクト \(F8h\), ページ 175](#) (ホストアプリケーションオブジェクト)、コマンド Pcess_CIP_Object_Request参照。

4.3 アセンブリマッピングオブジェクト (EBh) の使用

4.3.1 はじめに

本ガイドでは、アセンブリマッピングオブジェクト (EBh) を使用してCIPインスタンスをADIデータにマッピングする方法について説明します。

4.3.2 データの追加 - アプリケーションデータオブジェクト

Anybusオブジェクトモデルに従い、アプリケーションで使用されるすべてのデータはアプリケーションデータインスタンス (ADI) で表される必要があります。ADIはデータ構造体の一部で、各ADIは3種の可能なタイプ (変数、配列、または構造) のうち1つのみを表します。

詳細については、『ソフトウェアデザインガイド』の「アプリケーションデータオブジェクト (FEh)」を参照してください。

ADIが30個ある場合の例を以下に示します。インスタンス1~6 および30はアプリケーションに実装されていますが、7~29は実装されていないとします。

アプリケーションデータオブジェクト (FEh) インスタンス

インスタンス番号	実装状況	順番
1	あり	1
2	あり	2
3	あり	3
4	あり	4
5	あり	5
6	あり	6
7...29	なし	-
30	あり	7

4.3.3 データのグループ化 - アセンブリマッピングオブジェクト

アセンブリマッピングオブジェクトの使用により、任意の数のプロセスデータセットを作成 (いわゆるアセンブリマッピング) できます。各アセンブリマッピングインスタンスがプロセスデータの論理セットそれぞれを表します。これらはネットワーク側から選択され単一の接続で受信することができます。

以下に示すように、アセンブリマッピングオブジェクトの各インスタンスに、任意の数のADIを参照するADIマップが含まれています。

インスタンス番号は自由に設定できます。

アセンブリマッピングオブジェクト (EBh) インスタンス

インスタンス番号	タイプ	ADIマップ
1	Read	1、2
2	Read	2、3
10	Write	3、4、30
11	Write	4、5
30	Read	5、6
51	Write	6、30

アセンブリマッピングオブジェクトには、Write PD Instance ListとRead PD Instance Listという2つのオブジェクトインスタンスアトリビュートがあります。これら2つのアトリビュートにはそれぞれ、すべての読み出しインスタンスとすべての書き込みインスタンスへの参照が含まれます。上記の例では、これら2つのアトリビュート内に次のようにコンテンツが自動的に生成されます。

名前	アトリビュート	値
Write PD Instance List	11	10、11、51
Read PD Instance List	12	1、2、30




*Write PD Instance List*アトリビュートおよび*Read PD Instance List*アトリビュートはネットワークのビューを適用します。すなわち、入力ネットワーク上にデータを生成し、出力はネットワーク上のデータを消費します。

*Write PD Instance List*には、タイプが「Read」のアセンブリマッピングオブジェクトインスタンスがすべて含まれます。*Read PD Instance List*には、タイプが「Write」のアセンブリマッピングオブジェクトインスタンスがすべて含まれます。

4.3.4 CIPアセンブリ番号の設定

アセンブリマッピングオブジェクトのリードおよびライトインスタンスリストアトリビュートは、EtherNet/IPホストオブジェクト内の対応する2つのアトリビュートに、下表に示すように結び付けられます。

このようにCIPインスタンス番号をアセンブリマッピングオブジェクトインスタンスに関連付けることで、アプリケーションデータをCIPアセンブリデータにルーティングします。

 リストはインデックスに基づいてマッチングが行われるため、長さが同じでなければなりません。

アセンブリマッピングオブジェクトアトリビュート	値		値	EtherNet/IPホストオブジェクトインスタンスアトリビュート
11 - Write PD Instance List	10	<->	10	7 - Producing Instance Number
	11	<->	22	
	51	<->	100	
12 - Read PD Instance List	1	<->	1	8 - Consuming Instance Number
	2	<->	2	
	30	<->	150	



CIP仕様に準拠するためには、Write_Assembly_DataとRead_Assembly_Dataの両方のサービスを実装する必要があります。

4.3.5 その後

初期化フェーズのNW_INIT状態時に、すべての書き込みアセンブリ (「Write」型アセンブリマッピングオブジェクトのインスタンスなど) がライトプロセスデータ領域に再マッピングされます。これを実行するために、デバイスはホスト内のアプリケーションデータオブジェクトにRemap_ADI_Write_Areaコマンドを発行します。

詳細については、『Anybus CompactCom 40 Software Design Guide』の「Runtime Remapping of Process Data」付録を参照してください。

ネットワークが初期化されると、デバイスはNW_INITからWAIT_PROCESSの状態に移行します。デバイスがForward Open要求を受信すると、要求内のproducing/consumingパラメータがEtherNet/IPホストオブジェクトインスタンス番号(producing/consuming) に対して検証・照合されます。

検証に成功すると、リードプロセスデータがリマッピングされ、デバイスはPROCESS_ACTIVE状態に移行します。続いてI/O接続が確立され、ネットワーク上でのデータ交換が可能になります。

4.4 ソケットインターフェース (上級者対象)

内蔵ソケットインターフェースでTCP/IPの上に追加のプロトコルを実装できます。

下記も参照してください。

- [ソケットインターフェースオブジェクト \(07h\), ページ 127](#) (Anybusモジュールオブジェクト)
- [メッセージの分割, ページ 142](#)

4.5 診断

処理中の全イベントの重大度値が(論理ORを用いて)結合され、Identity オブジェクト (CIP) の「状態」アトリビュート内の対応ビットにコピーされます。

下記も参照してください。

- [Identity オブジェクト \(01h\), ページ 78](#) (CIPオブジェクト)
- [診断オブジェクト \(02h\), ページ 116](#) (Anybusモジュールオブジェクト)

4.6 QuickConnect

モジュールはQuickConnect機能に対応します。これはEtherNet/IPホストオブジェクトで有効化されています。このモジュールは、クラスAに準拠しています。パラレル、SPI、シフトレジスタによるアプリケーションインターフェースでマップされた16 バイトのI/O データで、起動時間が180 ms未満。

下記も参照してください。

- [EtherNet/IPホストオブジェクト \(F8h\), ページ 175](#) (ホストアプリケーションオブジェクト)
- [TCP/IPインターフェースオブジェクト \(F5h\), ページ 106](#) (CIPオブジェクト)

4.7 CIPセーフティ

Anybus CompactCom 40 EtherNet/IPはCIPセーフティ プロファイルをサポートしています。このプロファイルで、ブラックチャネルインターフェースからのデータ送信が可能になります。すなわち、IXXAT Safe T100などのアドオンセーフティモジュールを使って、EtherNet/IP上でセーフチャネルが確立されます。アプリケーションがCIPセーフティをサポートできるようにするには、ファンクショナルセーフティオブジェクト (E8h、ホストアプリケーションオブジェクト) を実装する必要があります。Anybus CompactCom シリアルチャネルがファンクショナルセーフティ通信に使用されます。このチャネルがホストアプリケーション用に使用される場合は、第二の別のシリアルチャネルがファンクショナルセーフティ通信に実装されます。詳細については、『Anybus CompactCom Hardware Design Guide』を参照してください。

以下を参照してください。

- [ファンクショナルセーフティモジュール・オブジェクト \(11h\), ページ 154](#)
- [ファンクショナルセーフティオブジェクト \(E8h\), ページ 163](#)

4.7.1 セーフティモジュールのファームウェアアップグレード

接続されたセーフティモジュールのファームウェアは、Anybus CompactComを介してアップグレードできます。セーフティファームウェア (hiff ファイル) は、Anybus CompactComのファームウェアディレクトリにダウンロードする必要があります。Anybus CompactComは、再起動時にファームウェアを検出して検証します。ファームウェアのアップグレードが進行中であることは、ファンクショナルセーフティオブジェクト (E8h) のアトリビュート#5 (インスタンス #1) がTRUEに設定されることによって、アプリケーションに示されます。Anybus CompactComは初期化にかなりの時間を必要とします。この間はモジュールを再起動しないでください。

4.7.2 ネットワークからのリセット要求

CIPセーフティが有効になっている場合、リセット要求がネットワークから到着した後、1秒の遅延を経てからAnybus CompactCom 40 EtherNet/IPがリセットされます。

4.8 CIP Sync

Common Industrial Protocol (CIP) の時刻同期技術はCIP Syncと呼ばれます。この技術により、タイムスタンプ、イベントシーケンスの記録、分散型モーションコントロール、コントロール調整機能の強化が必要な高度分散型アプリケーションのサポートなどに求められる、CIPネットワーク接続デバイスおよびコントローラの正確なリアルタイム同期が可能になります。CIP SyncはIEEE 1588 (IEC 61588) 規格 - ネットワーク計測および制御システムのための高精度クロック同期プロトコル - に基づいており、一般的にPrecision Time Protocol (PTP) と呼ばれます。このプロトコルはイーサネットのようなローカルエリアネットワーク用に設計されていますが、これらに限定されません。このプロトコルは、分散型デバイスのネットワーク全体に、クロック同期のための標準的なメカニズムを提供します。IEEE標準化組織委員会にIEEE 1588-2008規格をオーダーすると、同規格について記したドキュメントを入手することができます。Time Syncオブジェクト (43h, CIPオブジェクト) はIEEE 1588 (IEC 61588) 規格にCIPインターフェースを提供します。CIP SyncはEtherNet/IPホストオブジェクト (F8h) でアトリビュート#32を設定すると有効になります。

以下を参照してください。

- [EtherNet/IPホストオブジェクト \(F8h\), ページ 175](#)
- [Time Syncオブジェクト \(43h\), ページ 92](#) (CIPオブジェクト)

5 FTPサーバ

5.1 基本情報

内蔵FTPサーバで標準のFTPクライアントを使用してファイルシステムを容易に管理できます。FTPサーバはイーサネットホストオブジェクト (F9h) のアトリビュート#6を使用して無効化できます。

デフォルトでは以下のポート番号がFTP通信に使用されます:

- TCP、ポート20 (FTPデータ ポート)
- TCP、ポート21 (FTPコマンド ポート)

FTPサーバは二つの並列クライアントまで対応します。

5.2 ユーザーアカウント

ユーザーアカウントは設定ファイル \ftp.cfg に保存されます。このファイルに全ユーザーのユーザー名、パスワード、ホームディレクトリが保存されています。ユーザーはユーザー用ホームディレクトリの外部のファイルにはアクセスできません。

ファイルフォーマット:

```
User1:Password1:Homedirectory1
User2:Password2:Homedirectory2
User3:Password3:Homedirectory3
```

オプションとしては下記のように、ユーザーリストを含むファイルへのパスでUserN:PasswordNセクションを置換できます:

ファイルフォーマット (\ftp.cfg) :

```
User1:Password1:Homedirectory1
User2:Password2:Homedirectory2
.
.
UserN:PasswordN:HomedirectoryN
\path\userlistA:HomedirectoryA
\path\userlistB:HomedirectoryB
```

ユーザーリストを含むファイルのフォーマットは、次のとおりです。

ファイルフォーマット:

```
User1:Password1
User2:Password2
User3:Password3
.
.
.
UserN:PasswordN
```

注:

- ユーザー名は16文字を超えることはできません。
- パスワードは16文字を超えることはできません。
- ユーザー名とパスワードにはアルファベットのみ使用できます。
- \ftp.cfgがないか解釈されない場合、全てのユーザー名/パスワードの組合せが承認され、homeディレクトリはFTP root (\ftp\)になります。
- ユーザーログインが必要な場合はユーザーのhome ディレクトリもファイルシステムに存在しなければなりません。ユーザー情報をftp.cfgファイルに追加するだけでは不十分です。
- AdminモードがEthernetオブジェクトで有効にされている場合、どのユーザー名/パスワードの組合せでも承認され、ユーザーはファイルシステムに自由にアクセスできます (つまり、home ディレクトリがシステムrootになる)。vfs フォルダは読み取り専用です。
- rootアクセス(\)権限をもつユーザーを最低でも一人作成することを強く推奨します。そうでない場合、システムファイルを変更する必要があるたびにAdmin モードを有効にしなければなりません (\ftp.cfgを含む)。

5.3 セッション例

Windowsエクスプローラにはファイルシステムに以下の手順でアクセスするために使い易い内蔵FTPクライアントがあります:

1. Windowsエクスプローラを開きます。
2. アドレスバーにFTP://<user>:<password>@<address>とタイプします
 - <address>の代わりにAnybus モジュールのIPアドレスを記入します
 - <user>の代わりにユーザー名を記入します
 - <password>の代わりにパスワードを入力します
3. **Enter**を押します。Explorerは指定された設定を使用してAnybusモジュールに接続を試みます。接続に成功すると、Explorerウィンドウにファイルシステムが表示されます。

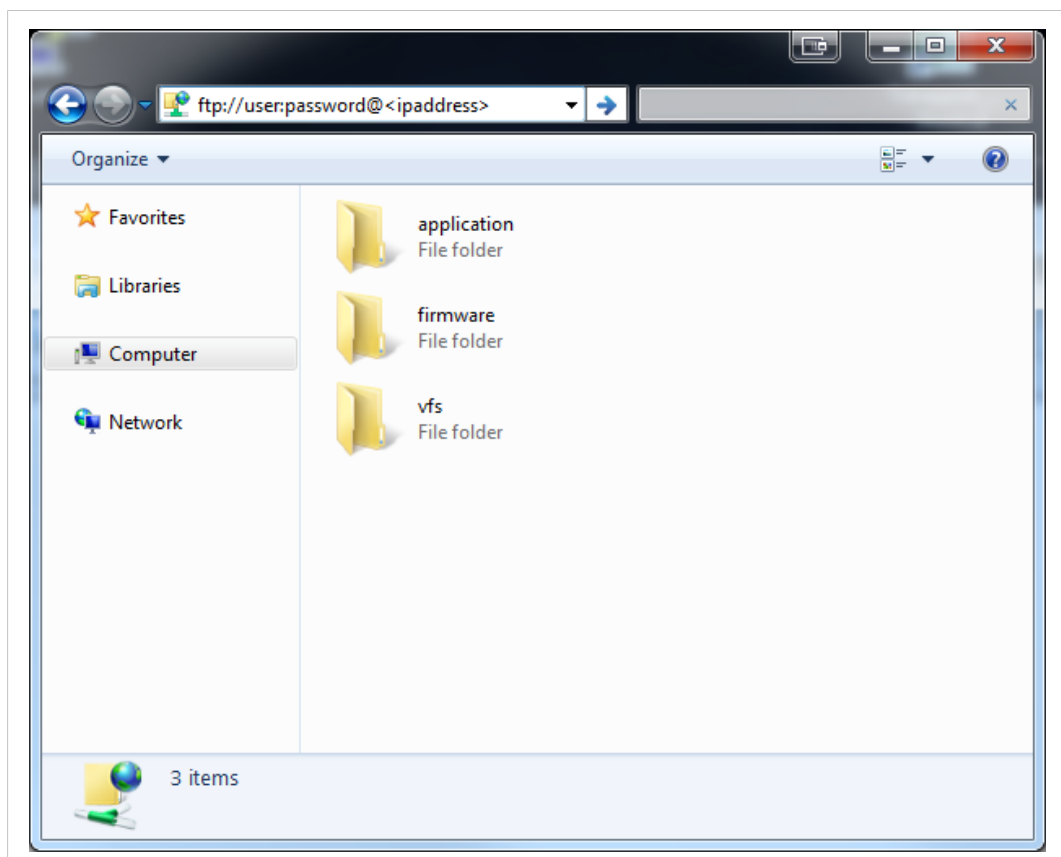


図 3

6 Webサーバ

6.1 基本情報

内蔵Webサーバはエンドユーザーのインタラクションや構成のためにフレキシブルな環境を提供します。JSON、SSI、クライアントサイドスクリプトはオブジェクトやファイルシステムデータへのアクセスを可能にし、高度なGUIを作成できるようになります。

Webインターフェースはファイルシステムに保存され、これにはFTPサーバからアクセスできます。必要であれば、Webサーバは Ethernet ホストオブジェクト (F9h) で完全に無効にできます。

下記も参照してください。

- [FTPサーバ, ページ 24](#)
- [Server Side Include \(SSI\), ページ 36](#)
- [JSON, ページ 56](#)
- [Ethernet ホストオブジェクト \(F9h\), ページ 184](#)
- [ネットワークコンフィグレーションオブジェクト \(04h\), ページ 118](#)

6.2 デフォルト Web ページ

デフォルト Web ページから以下の項目へアクセスできます:

- ネットワークコンフィグレーションパラメータ
- ネットワークの状態情報
- ホストアプリケーションADIへのアクセス

デフォルト Web ページは、vfs フォルダからアクセス可能な仮想ファイルシステムに保存されているファイルから構成されています。これらのファイルは読取り専用であり、削除も上書もできません。Webサーバはまず Webのrootフォルダにあるファイルを検索します。ここに見つからない場合は vfs フォルダにあるファイルを検索し、ファイルがあたかもWebの root フォルダにあるように表示します。vfs フォルダのデフォルトファイルと全く同じ名前でWebのroot フォルダにファイルを読み込むと、ウェブページをカスタム化することができ、写真、ロゴやスタイルシートなどを置き換えることができます。

完全にカスタム化されたWebシステムが設計され、vfsフォルダのファイルを全く使用しない場合、仮想ファイルシステムを無効にすることが推奨されます。ファイルシステム インターフェースオブジェクトを参照してください。

下記も参照してください。

- [ファイルシステム, ページ 16](#)
- [Anybusファイルシステムインターフェースオブジェクト \(0Ah\), ページ 149](#)

6.2.1 ネットワークコンフィグレーション

ネットワークコンフィグレーションページは、ネットワークコンフィグレーションオブジェクトにあるTCP/IPとSMTP設定を変更するためのインターフェースを提供します。

The screenshot displays the Anybus CompactCom web interface for network configuration. The browser address bar shows the URL 10.11.21.232. The user is logged in as 'MyAccount (administrator)'. The interface is divided into several sections:

- IP Configuration:**
 - Overview: DHCP is set to Disabled.
 - Parameters: IP Address is 10.11.21.232, Subnet Mask is 255.255.255.0.
 - NETWORK: Gateway Address is 0.0.0.0.
 - Status: Host Name is empty.
 - Configuration: Domain name is empty.
 - SERVICES: DNS Server #1 and #2 are both 0.0.0.0.
 - SMTP: DNS Server #1 and #2 are both 0.0.0.0.
 - SHICP: No settings.
 - SECURITY: No settings.
 - Certificates: No settings.
 - Accounts: No settings.
- Ethernet Configuration:**
 - Port 1: Auto
 - Port 2: Auto
- MQTT Configuration:**
 - Broker URL: Empty
 - TLS: Enable
 - Client Identifier: Empty
 - Keep alive time (s): 60
 - Base topic: Empty
 - Quality of service: QoS 0
- MQTT Broker Account Configuration:**
 - Username: Empty
 - Password: Empty
- OPC UA Configuration:**
 - TCP port: 4840
 - Discovery server URL: Empty
 - SecurityPolicyNone: Disable

Each configuration section includes a 'Save settings' button. The footer contains the copyright notice '© 2019 HMS Industrial Networks - All rights reserved' and the logo 'Connecting Devices™'.

図 4

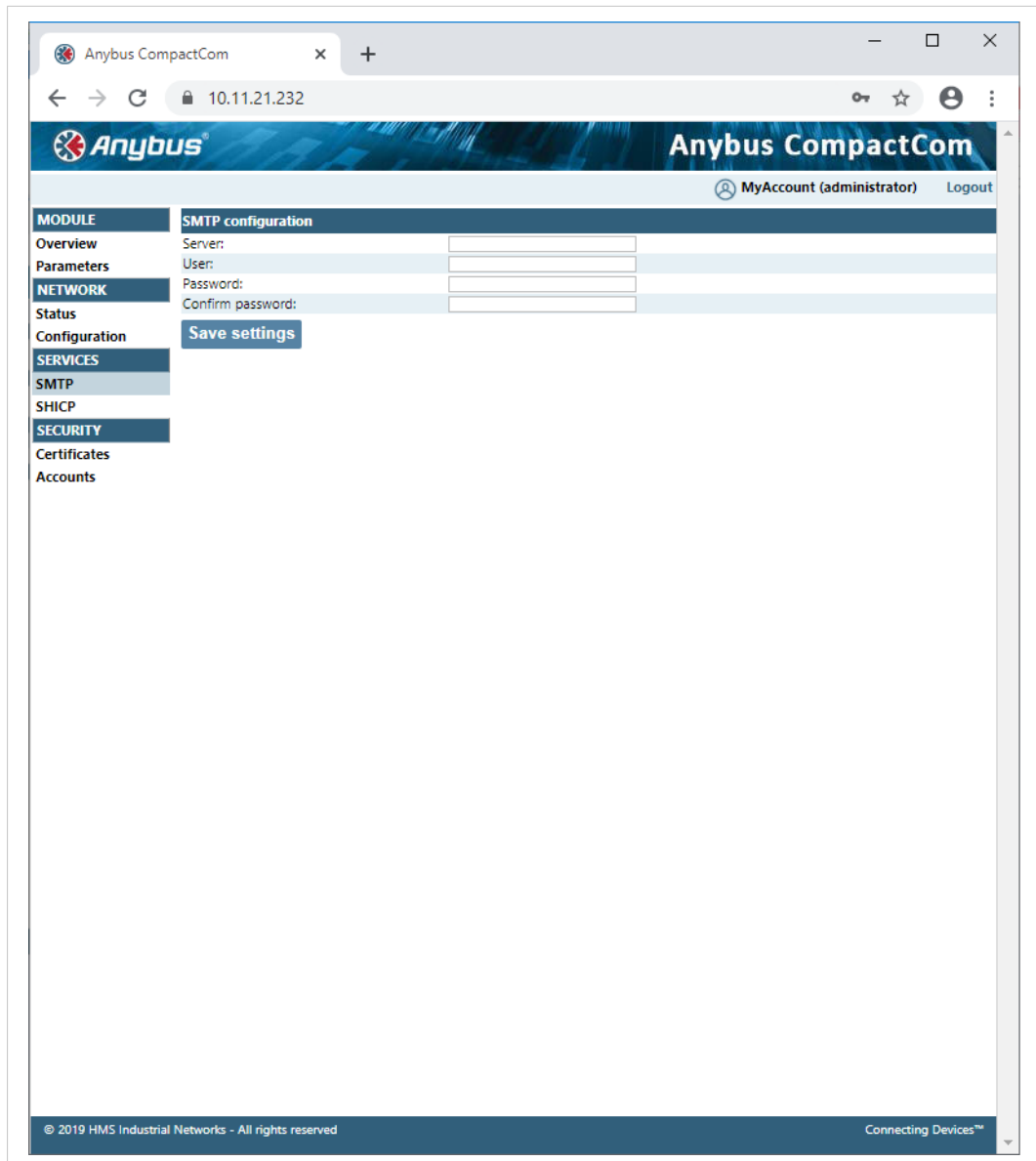


図 5

TCP/IPとSMTP設定を有効にするには、モジュールのリセットが必要です。イーサネットコンフィグレーションの設定は直ちに有効になります。

IP設定

モジュールはあらゆる変更を適用するためにリセットする必要があります。

名前	説明
DHCP	DHCP有効化/無効化 デフォルト値: 有効
IP address	モジュールのTCP/IP設定 デフォルト値: 0.0.0.0値の範囲: 0.0.0.0~255.255.255.255
Subnet mask	
Gateway	
Host name	IPアドレスまたは名前 最大64文字
Domain name	IPアドレスまたは名前 最大48文字
DNS 1	プライマリとセカンダリDNSサーバ (ホスト名解決に使用) デフォルト値: 0.0.0.0値の範囲: 0.0.0.0~255.255.255.255
DNS 2	

Ethernet設定

変更は直ちに適用されます。

名前	説明
Port 1	Ethernet speed/duplex設定 デフォルト値: auto
Port 2	

SMTP設定

あらゆる変更は適用される前にモジュールのリセットが必要です。

名前	説明
Server	IPアドレスまたは名前 最大64文字
User	最大64文字
Password	最大64文字
Confirm password	

6.2.2 Ethernet 統計ページ

Ethernet 統計Webページは以下の情報を含みます:

Ethernetリンク		説明
Port 1	Speed:	現在のリンクスピード。
	Duplex:	現在のduplex設定。
Port 2	Speed:	現在のリンクスピード。
	Duplex:	現在のduplex設定。

Ethernet/IP統計		説明
Established Class1 Connections		現在確立済みのクラス1コネクション数
Established Class3 Connections		現在確立済みのクラス3コネクション数
Connection Open Requests		受信された接続開始要求数
Connection Open Format Rejects		要求フォーマットエラーにより拒否された接続開始要求数
Connection Open Resource Rejects		リソース不足により拒否された接続開始要求数
Connection Open Other Rejects		その他の理由により拒否された接続開始要求数
Connection Close Requests		受信された接続開始要求数
Connection Close Format Rejects		要求フォーマットエラーにより拒否された切断要求数
Connection Close Other Rejects		その他の理由により拒否された切断要求
Connection Timeouts		接続タイムアウト回数

Interface Counters	説明
In Octets:	受信バイト数。
In Ucast Packets:	受信ユニキャストパケット数。
In NUcast packets:	受信非ユニキャストパケット数 (ブロードキャストとマルチキャスト)。
In Discards:	利用可能なメモリーバッファが無い為に棄却された受信パケット数
In Errors:	受信エラーにより棄却された受信パケット数。
In Unknown Protos:	未対応のプロトコルタイプで受信されたパケット数。
Out Octets:	送信済みバイト数。
Out Ucast packets:	送信済みユニキャストパケット数。
Out NUcast packets:	送信済み非ユニキャストパケット数 (ブロードキャストとマルチキャスト)。
Out Discards:	利用可能なメモリーバッファが無い為に棄却された送信パケット数。
Out Errors:	伝送エラー。

Media Counters	説明
Alignment Errors	長さのオクテットが整数になっていない受信フレーム数。
FCS Errors	FCSチェックに失格した受信フレーム数。
Single Collisions	一回だけコリジョンが発生して転送に成功したフレーム数。
Multiple Collisions	二回以上コリジョンが発生して転送に成功したフレーム数。
SQE Test Errors	SQEテストエラーメッセージが生成された回数。 (現在のPHYインターフェースでは提供されていません。)
Deferred Transmissions	第一回の伝送試行が媒体ビジーにより遅延しているフレーム数。
Late Collisions	パケット伝送に対して512ビット目より後でコリジョンが検出された回数。
Excessive Collisions	コリジョンが多すぎて伝送に失敗したフレーム数。
MAC Receive Errors	インターフェースの受信が内部MACのサブレイヤー受信エラーにより失敗したフレーム数。
MAC Transmit Errors	内部MACのサブレイヤー受信エラーにより伝送が失敗したフレーム数。
Carrier Sense Errors	搬送波検出条件が損失したまたはフレーム転送の試行時にアサートされなかった回数。
Frame Size Too Long	最大許容フレームサイズを超過した受信フレーム数。
Frame Size Too Short	最小許容フレームサイズより小さいサイズの受信フレーム数。

CIP Syncステータス	説明
Grandmaster clock identity	Time Syncオブジェクト (43h) , ページ 92の属性#8のClockIdentityと同等。
Parent clock identity	Time Syncオブジェクト (43h) , ページ 92の属性#9のClockIdentityと同等。
Local clock identity	Time Syncオブジェクト (43h) , ページ 92の属性#10のClockIdentityと同等。
Grandmaster clock class	Time Syncオブジェクト (43h) , ページ 92の属性#8のClockClassと同等。
Local clock class	Time Syncオブジェクト (43h) , ページ 92の属性#10のClockClassと同等。
Synchronization status	値の文字列挙型。 Time Syncオブジェクト (43h) , ページ 92の属性#2と同等。
Current offset to master [ns]	Time Syncオブジェクト (43h) , ページ 92の属性#5と同等。
Current port status	値の文字列挙型。 Time Syncオブジェクト (43h) , ページ 92の属性#12のPortStateと同等。

6.3 サーバコンフィグレーション

6.3.1 基本情報

Webサーバコンフィグレーションの基本設定はシステムファイル\http.cfgに保存されます。このファイルには、Webサーバ名、Webインターフェース用ルートディレクトリ、コンテンツタイプ、SSI用にスキャンされるべきファイルタイプ一覧が保存されています。

```
File Format:
[ServerName]
WebServerName
[WebRoot]
\web

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

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

Webサーバ名 [ServerName]	モジュールによる応答のHTTPヘッダに含まれるWebサーバ名を設定します。
Webルートディレクトリ [WebRoot]	Webサーバは、このディレクトリの外にあるファイルにはアクセスできません。
コンテンツタイプ [FileTypes]	ファイル拡張子とその報告済みコンテンツタイプのリスト。 下記も参照してください。 Default content types 項
SSIファイルタイプ [SSIFileTypes]	デフォルトでは、拡張子が「shtml」のファイルだけがSSI用にスキャンされます。 必要に応じて、追加のSSI ファイルタイプをここで追加できます。

Webのルートディレクトリがウェブインターフェース関連のファイルの場所を決定します。このディレクトリとそのサブディレクトリ以外にあるファイルは、Webサーバからアクセスすることができません。

6.3.2 インデックスページ

モジュールは以下のフォルダーにインデックスページがないか検索します：

1. <WebRoot>\index.htm
2. <WebRoot>\index.html
3. <WebRoot>\index.shtml
4. <WebRoot>\index.wml



<WebRoot>を `\http.cfg`で規定のWebルート ディレクトリで置き換えます。

インデックスページが見つからなければ、モジュールは仮想インデックスファイル (有効な場合)をデフォルトの場所に指定します。

下記も参照してください。

- デフォルト Webページ

6.3.3 デフォルト コンテンツタイプ

デフォルトでは以下のコンテンツタイプをファイル拡張子で識別できます。

ファイル拡張子	報告されたコンテンツタイプ
htm, html, shtm	text/html
gif	image/gif
jpeg, jpg, jpe	image/jpeg
png	image/x-png
js	application/x-javascript
bat, txt, c, h, cpp, hpp	text/plain
zip	application/x-zip-compressed
exe, com	application/octet-stream
wml	text/vnd.wap.wml
wmlc	application/vnd.wap.wmlc
wbmp	image/vnd.wap.wbmp
wmls	text/vnd.wap.wmlscript
wmlsc	application/vnd.wap.wmlscriptc
xml	text/xml
pdf	application/pdf
css	text/css

コンテンツタイプはサーバ設定ファイルに追加することによって追加または再定義することができます。

6.3.4 認証

ディレクトリは、ディレクトリ内に「web_accs.cfg」という保護用ファイルを置くことで、Webアクセスから保護することができます。このファイルに、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ているユーザー一覧が含まれています。

オプションとして、[AuthName]キーを含めるとログインメッセージを指定することができます。このメッセージは保護されたディレクトリにアクセスするとWebブラウザに表示されます。

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

[AuthName]
(message goes here)
```

許可されたユーザーの一覧はオプションとしてその他の一つまたは複数のファイルへリダイレクト可能です。



許可されたユーザーの一覧がその他のファイルに置かれたときは、このファイルにネットワークからアクセスし読み取れることに注意してください。

次の例では、許可されたユーザーの一覧がhere.cfgとtoo.cfgから読み込まれます。

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

[AuthType]
Basic

[AuthName]
Howdy. Password, please.
```

「AuthType」フィールドは、認証スキームを識別するために使用されます。

値	説明
基本	ブレンテキストのパスワードを使用したWeb認証方法。
Digest (ダイジェスト)	challenge-response認証によるより安全性の高い方法。[AuthType]欄が指定されているときにはデフォルトで使用される。

7 電子メールクライアント

7.1 基本情報

内蔵の電子メールクライアントは、アプリケーションにSMTPサーバを通して電子メールメッセージを送信する事を許可します。メッセージはSMTPクライアントオブジェクト (04h)で直接指定またはファイルシステムから取得可能です。後者はSSIを含むことができますが、技術的な理由で一部のコマンドは使用できません (各SSIコマンド用に個別指定)。

クライアントは'LOGIN'メソッドによる認証に対応します。アカウント設定などはネットワークコンフィグレーションオブジェクト (04h)に保存されます。

7.2 電子メールメッセージを送る方法

電子メールメッセージを送るにはSMTPアカウントを設定しなければなりません。

その内容には以下が含まれます：

- 有効なSMTPサーバアドレス
- 有効なユーザー名
- 有効なパスワード

電子メールメッセージは次の手順で送ります：

1. Createコマンド(03h)を使って新たな電子メールインスタンスを作成します
2. 送信者、受信者、件名、メッセージ本文を電子メールインスタンスに指定します
3. Send Instance Emailコマンド (10h)を電子メールインスタンスに対して発行します
4. オプションとして、Deleteコマンド (04h)を使って電子メールインスタンスを削除します

Send Email from Fileコマンドを使ってファイルシステムにあるファイルに基づいてメッセージを送ります。このコマンドはSMTPクライアントオブジェクト (04h)に記述されています。

8 Server Side Include (SSI)

8.1 基本情報

Server Side Include機能 (SSI) でファイルやオブジェクトからのデータをウェブページおよび電子メールメッセージに表わすことができます。

SSIはソースドキュメントの中に埋め込まれた特殊なコマンドです。Anybus CompactCom モジュールにこうしたコマンドが適用されると、それを実行し、結果で (適用される限り) 置換します。

デフォルトでは拡張子 'shtm' のファイルだけがSSI用にスキャンされます。

8.2 Include File

この機能はファイルの内容を含みます。内容はSSIのためにスキャンされます。



この機能は電子メールメッセージで使用できません。

構文:

```
<?--#include file="filename"-->
```

filename: ソースファイル

シナリオ	デフォルト出力
成功	(ファイルの内容)

8.3 コマンドの機能

8.3.1 基本情報

コマンドの機能はコマンドを実行しその結果を含みます。

基本構文

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

command: コマンドの機能については下記参照



「コマンド」は最大500文字に制限されています。

コマンドの機能

コマンド	電子メールメッセージに関して有効
GetConfigItem()	Yes
SetConfigItem()	No
SsiOutput()	Yes
DisplayRemoteUser	No
ChangeLanguage()	No
IncludeFile()	Yes
SaveDataToFile()	No
printf()	Yes
scanf()	No

8.3.2 GetConfigItem()

このコマンドはファイルシステムにあるファイルから特定の情報を返します。

ファイルフォーマット

ソースファイルは次の形式に従わなければなりません:

```
[key1]
value1

[key2]
value2
...
[keyN]
valueN
```

構文:

```
<?--exec cmd_argument='GetConfigItem("filename", "key"[,"separator"])'-->
```

filename: 読み出し元のソースファイル
 key: ファイルのソース[key]
 separator: オプション。行区切り文字 ("
"など) を指定します。
 (デフォルトは CRLF)

デフォルト出力

シナリオ	デフォルト出力
成功	(指定されたキーの値)
認証エラー	「Authentication error」
ファイルオープンエラー	「Failed to open file 『ファイル名』」
キーが見つからない	「Tag (key) not found」

例

次のSSIの場合...

```
<?--exec cmd_argument='GetConfigItem("\example.cnf", "B")'-->
```


... をファイル ('example.cnf') と併用すると...

```
[A]  
First  
[B]  
Second  
[C]  
Third
```

... 文字列 'Third'が返ります。

8.3.3 SetConfigItem()

この関数はHTMLフォームをファイルとしてファイルシステムに保存します。

 この機能は電子メールメッセージで使用できません。

ファイルフォーマット


各フォームオブジェクトは[tag]として格納され、その後実際の値が続きます。

```
[form object name 1]
form object value 1

[form object name 2]
form object value 2

[form object name 3]
form object value 3

...
[form object name N]
form object value N
```

 アンダースコアで始まる名前のフォームオブジェクトは保存されません。

構文:

```
<?--exec cmd_argument='SetConfigItem("filename"[, Overwrite])'-->
```

filename: ターゲットファイル。指定されたファイルが存在しない場合、作成されます (パスが有効であることが前提)。

Overwrite: オプション。コマンドが発行されるたびにモジュールは新しいファイルを作成します。デフォルト機能は既存のファイルを変更します。

デフォルト出力

シナリオ	デフォルト出力
成功	「Configuration stored to 『ファイル名』」
認証エラー	「Authentication error」
ファイルオープンエラー	「Failed to open file 『ファイル名』」
ファイル書き込みエラー	「Could not store configuration to 『ファイル名』」

例

次の例はこの機能の使い方を示しています。結果のページは自身宛にフォームを送信し、SetConfigItem コマンドで評価されます。

```
<HTML>
  <HEAD><TITLE>SetConfigItem Test</TITLE></HEAD>
  <BODY> <!--#exec cmd_argument='SetConfigItem("\food.txt")'-->

  <FORM action="test.shtm">
  <P>
  <LABEL for="Name">Name: </LABEL><BR>
  <INPUT type="text" name="Name"><BR><BR>

  <LABEL for="_Age">Age: </LABEL><BR>
  <INPUT type="text" name="_Age"><BR><BR>

  <LABEL for="Food">Food: </LABEL><BR>
  <INPUT type="radio" name="Food" value="Cheese"> Cheese<BR>
  <INPUT type="radio" name="Food" value="Sausage"> Sausage<BR><BR>

  <LABEL for="Drink">Drink: </LABEL><BR>
  <INPUT type="radio" name="Drink" value="Wine"> Wine<BR>
  <INPUT type="radio" name="Drink" value="Beer"> Beer<BR><BR>

  <INPUT type="submit" name="_submit">
  <INPUT type="reset" name="_reset">
  </P>
  </FORM>


</BODY>
</HTML>
```

出力されるファイル ('food.txt') はおよそ次のようになります。

```
[Name]
Cliff Barnes

[Food]
Cheese

[Drink]
Beer
```

 この例が機能するには、HTMLファイルに「test.shtm」という名前を付ける必要があります。

8.3.4 SsiOutput()

このコマンドは以下のコマンド関数のSSI出力を一時的に変更します。

構文:

```
<?--#exec cmd_argument='SsiOutput("success", "failure")'-->
```

success: 成功したときに使用する文字列

failure: 失敗したときに使用する文字列

デフォルト出力

(このコマンド自体は出力がありません)

例

次の例はこのコマンドの使い方を示します。


```
<?--#exec cmd_argument='SsiOutput ("Parameter stored", "Error")'-->
<?--#exec cmd_argument='SetConfigItem("File.cfg", Overwrite)'-->
```

下記も参照してください。

- [SSI出力の設定, ページ 55](#)

8.3.5 DisplayRemoteUser

このコマンドは認証セッションにおけるユーザー名を返します。

 このコマンドは電子メールメッセージでは使用できません。

構文:

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

デフォルト出力

シナリオ	デフォルト出力
成功	(現在のユーザー)

8.3.7 IncludeFile()

このコマンドはファイルの内容を含みます。その内容がSSIに関してはスキャンされないことに注意してください。

構文:

```
<?--#exec cmd_argument='IncludeFile("filename" [, separator])'-->
```

filename: ソースファイル
separator: オプション。行区切り文字 (「
」など) を指定します。

デフォルト出力

シナリオ	デフォルト出力
成功	(ファイル内容)
認証エラー	「Authentication error」
ファイルオープンエラー	「Failed to open file 『ファイル名』」

例

次の例はこの機能の使い方を示しています。

```
<HTML>
<HEAD><TITLE>IncludeFile Test</TITLE></HEAD>
<BODY>
<H1> Contents of 'info.txt':</H1>
<P>
<?--#exec cmd_argument='IncludeFile("info.txt")'-->.
</P>
</BODY>
</HTML>
```

‘info.txt’の内容:

```
Neque porro quisquam est qui dolorem ipsum quia dolor sit
amet,consectetur, adipisci velit...
```

ブラウザで見ると結果ページはだいたい次のようになります：

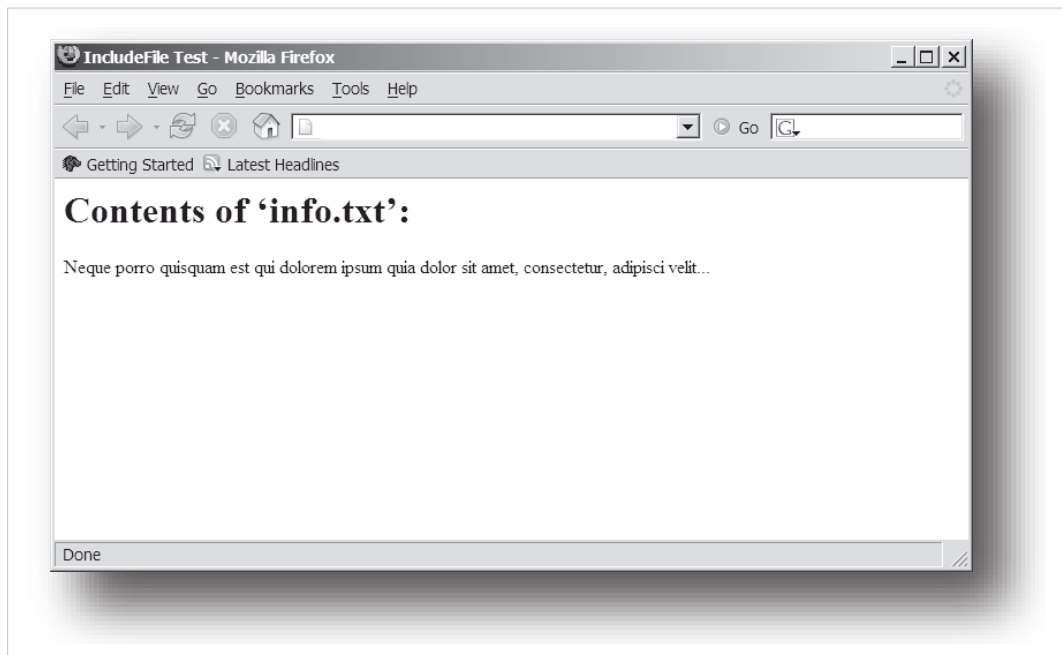


図 6

下記も参照してください。

- [Include File, ページ 36](#)

8.3.8 SaveDataToFile()

このコマンドはHTMLフォームデータをファイルとしてファイルシステムに保存します。異なるフォームオブジェクトの内容は空白行 (2*CRLF) で区切ります。



この機能は電子メールメッセージで使用できません。

構文:

```
<!--#exec cmd_argument='SaveDataToFile("filename" [, "source"],
Overwrite|Append)'-->
```

filename	ターゲットファイル。指定されたファイルが存在しない場合、作成されます (パスが有効であることが前提)。
source:	オプション。フォームオブジェクトを指定すると、その特定のフォームオブジェクトデータのみが保存されます。デフォルト動作は、名前がアンダースコアで始まるもの以外すべてのフォームオブジェクトデータを保存します。
Overwrite Append	データを既存のファイルに上書するか挿入するかを指定します。

デフォルト出力

シナリオ	デフォルト出力
成功	「Configuration stored to 『ファイル名』」
認証エラー	「Authentication error」
ファイル書き込みエラー	「Could not store configuration to 『ファイル名』」

例

次の例はこの機能の使い方を示しています。結果のページは自身宛にフォームを送信し、SaveDataToFile コマンドで評価されます。

```
<HTML>
<HEAD><TITLE>SaveDataToFile Test</TITLE></HEAD>
<BODY>

<?--#exec cmd_argument='SaveDataToFile("\stuff.txt", "Meat", Overwrite)'-->

<FORM action="test.shtm">
<P>
<LABEL for="Fruit">Fruit: </LABEL><BR>
<INPUT type="text" name="Fruit"><BR><BR>

<LABEL for="Meat">Meat: </LABEL><BR>
<INPUT type="text" name="Meat"><BR><BR>

<LABEL for="Meat">Bread: </LABEL><BR>
<INPUT type="text" name="Bread"><BR><BR>

<INPUT type="submit" name="_submit">
</P>
</FORM>

</BODY>
</HTML>
```

結果のファイル (\stuff.txt) には、「Meat」と名付けられているフォームオブジェクトに対して指定された値が含まれます。



この例が機能するには、HTMLファイルに「test.shtm」という名前を付ける必要があります。

8.3.9 printf()

この関数はAnybus CompactCom モジュールやアプリケーションのデータを含む場合もあるフォーマットされた文字列を返します。ここで使用されるフォーマット構文は標準のC関数printf()のものと同じです。

この関数はゼロまたは複数のフォーマットタグに数個の引数が続くかたちのテンプレート文字列を受け入れます。各フォーマットタグは単一の引数に対応し、その引数が人間の読めるかたちに変換される方法を決定します。

構文:

```
<!--#exec cmd_argument='printf("template" [,
argument1, ..., argumentN] )'-->
```

template:	引数を表す方法を決めるテンプレート。任意の数のフォーマットタグを含むことができ、これらに続く引数が代入されて、要求に従ってフォーマットされます。フォーマットタグの数は引数の数と一致していなければなりません。そうでない場合、結果は未定義です。 詳細については、後述の「フォーマットタグ」のセクションを参照してください。
argument:	ソースの引数。テンプレート文字列に挿入されるデータの実際のソースを指定するオプションのパラメータ。引数の数はフォーマットタグの数と同じでなければならず、これが違えば結果は未定義です。本文書の作成時点において可能な唯一の引数はABCCMessage()です。 下記も参照してください。 <ul style="list-style-type: none"> • ABCCMessage(), ページ 51

デフォルト出力

シナリオ	デフォルト出力
成功	(printf() 結果)
ABCCMessageエラー	ABCCMessageエラーの文字列 (エラー , ページ 54)

例

以下を参照してください。

- [ABCCMessage\(\)](#), ページ 51
- [例 \(Get_Attribute\);](#), ページ 53

フォーマットタグ

フォーマットタグの構文は次のようになります:

```
%[Flags][Width][.Precision][Modifier]type
```

- Type (必須)

Typeの文字は必須で、下記のように基本的な表現が決まります:

typeの文字	表現	例
c	単一文字	b
d, i	符号つき10進整数。	565
e, E	指数を表す浮動小数。	5.6538e2
f	通常の固定小数点による浮動小数。	565.38
g, G	指数が-4より小さいか精度以上であれば%eまたは%Eが使用され、その他の場合は%fが使用されます。後続するゼロ/小数点は印刷されません。	565.38
o	符号なしの八進表記	1065
s	文字列	Text
u	符号なしの10進整数	4242
x, X	16進整数	4e7f
%	文字としての%であり割当なし	%

- Flag (オプション)

Flag文字	意味
-	与えられた幅以内で結果を左揃え表示します (デフォルトは右揃え)
+	数値が正か負かを示すために、常に+または-を含めます。
(空白)	ある数値が+または-で始まらないときは空白文字が使われます。
0 (ゼロ)	フィールドに空白ではなくゼロを埋めてパディングします
#	%e、%E、%fに関して小数がなくても小数点を付けます。%xと%Xに関してそれぞれ0xまたは0Xの接頭文字が付きまます。

- Width (幅) (オプション)

Width	意味
数字	印刷される最小文字数を指定します。 印刷される値がこの数より小さいと結果はフィールドの幅が埋まるまで上記の内指定文字で埋められます。 結果が大きくても、結果が切り捨てられることはありません。

- Precision (精度) (オプション)

この欄の正確な意味はtype文字によって異なります:

typeの文字	意味
d, i, o, u, x, X	印刷すべき10進の最小桁数を指定します。印刷すべき値がこの数より短い場合、結果の残り部分は空白で埋められます。結果がより大きい数であっても結果は途中で分断されないことに注意してください。
e, E, f	小数点の後に印刷される桁数(デフォルトは6)を指定します。
g, G	印刷すべき有意な数の最大数を指定します。
s	印刷される最大文字数を指定します。
c	(影響なし)

- Modifier (修飾子)

Modifier文字	意味
hh	引数はSINT8かUINT8として解釈されます
h	引数はSINT16かUINT16として解釈されます
L	引数はSINT32かUINT32として解釈されます

8.3.10 scanf()

この関数は上記の printf() とよく似ていますが、出力ではなく入力のために使用される点が異なります。この関数はHTMLフォームオブジェクトから渡された文字列を読み取り、テンプレート文字列によって指定されたものとしてこれをパースし、その結果データを指定引数に送ります。ここで使用されるフォーマット構文は標準C関数のscanf()と同じです。

関数はソース、ゼロまたはそれ以上のフォーマットタグを含むテンプレート文字列、これに続く数個の引数を受け付けます。各引数は一個のフォーマットタグに対応し、これはHTMLフォームから読み取られたデータがターゲット引数に送られる前に解釈される方法を決定します。



このコマンドは電子メールメッセージでは使用できません。

構文:

```
<?--#exec cmd_argument='scanf("source", "template" [,
argument1, ..., argumentN])'-->
```

source	文字列を抽出する元のHTML フォームオブジェクトの名前。
template:	構文解析とデータの解釈方法を指定するテンプレート。データを続く引数へ送る前に変換方法を決定する任意の数のフォーマットタグを含むことができます。フォーマットタグの数は引数の数と同じでなければならず、そうでない場合の結果は未定義です。 詳細については、後述の「フォーマットタグ」のセクションを参照してください。
argument:	解釈されたデータの送り先を指定するディスティネーション引数。引数の数はフォーマットタグの数と同じでなければならず、これが違うと結果は未定義です。 本文書の作成時点において可能な唯一の引数はABCCMessage()です。 下記も参照してください。 <ul style="list-style-type: none"> ABCCMessage(), ページ 51

デフォルト出力

シナリオ	デフォルト出力
成功	「成功」
構文解析エラー	「Incorrect data format」
引数のデータ過多	「Too much data」
ABCCMessageエラー	ABCCMessageエラーの文字列 (エラー, ページ 54)

例

下記も参照してください。

[ABCCMessage\(\), ページ 51](#)

[例 \(Set_Attribute\);, ページ 53](#)

フォーマットタグ

フォーマットタグの構文は次のようになります:

```
%[*] [Width] [Modifier] type
```

- Type (必須)

Typeの文字は必須で、下記のように基本的な表現が決まります：

タイプ	入力	引数データ型
c	単一文字	CHAR
d	符号つき10進整数を受け入れる	SINT8 SINT16 SINT32
i	符号つきまたは符号なし10進整数を受け入れます。入力データの頭文字により、10進数、16進数、8進数として付与可能: 頭文字: 形式: 0x: 16進数 0: 8進数 1~9: 10進数	SINT8/UINT8 SINT16/UINT16 SINT32/UINT32
u	符号なし10進整数を受け入れます。	UINT8 UINT16 UINT32
o	オプションで符号つき8進整数を受け入れます。	SINT8/UINT8 SINT16/UINT16 SINT32/UINT32
x, X	オプションで符号つき16進整数を受け入れます。	SINT8/UINT8 SINT16/UINT16 SINT32/UINT32
e, E, f, g, G	オプションで符号つき浮動小数を受け入れます。浮動小数の入力フォーマットはオプションの特性を伴う数の文字列です: - 符号つきの値でも可 - 「E」または「e」とそれに続く整数で構成される、指数フィールドが後に続く10進の有理数を含んでいる、指数値にすることができます。	FLOAT
n	入力を消費しません。対応する引数は、scanfがオブジェクト入力から読み取った文字数を書き込む整数です。	SINT8/UINT8 SINT16/UINT16 SINT32/UINT32
s	空白なしの文字のシーケンスを受け入れる	STRING
[scanset]	スキャンリストによって指定された期待されたバイトセットから一連の空白文字以外の文字を受け入れます (例: '[0123456789ABCDEF]'). ' 'という括弧文字をこのセットの頭文字に指定することができます。最初の '[' の直後にあるキャレット文字 (^) は、スキャンリストを反転します。つまり、リストされているもの以外のすべての文字を許可します。	STRING
%	この時点での単一の%入力を受け入れます。割当や変換はされません。完全な変換仕様は%%である必要があります。	-

- * (任意)

データは読み取られても無視されます。対応する引数に代入されません。

- Width (幅) (オプション)

読み取られる最大文字数を指定します。

- Modifier (修飾子) (オプション)

別のデータサイズを指定します。

Modifier (修飾子)	意味
h	SINT8、SINT16、UINT8、またはUINT16
l	SINT32かUINT32

8.4 引数関数

8.4.1 基本情報

引数関数は特定のコマンド機能に対するパラメータとして供給されます。

基本構文:

(構文は状況により異なる)

引数関数:

関数	説明
ABCCMessage()	-

8.4.2 ABCCMessage()

この関数はモジュールまたはホストアプリケーション内のオブジェクトに対してオブジェクトを要求します。

構文

```
ABCCMessage(object, instance, command, ce0, ce1, msgdata, c_type, r_type)
```

object	ターゲットオブジェクトを指定します。
instance	ターゲットインスタンスを指定します。
command	コマンド番号を指定します。
ce0	コマンドメッセージ用のCmdExt[0]を指定します。
ce1	コマンドメッセージ用のCmdExt[1]を指定します。
msgdata	コマンド内のMsgData[]サブフィールドの実際の内容を指定します。 <ul style="list-style-type: none"> データは直接形式で提供できます (形式はc_typeによって異なります) キーワード「ARG」は、データが親コマンド (scanf()など) から供給される際に使用されます。
c_type:	コマンド (msgdata) におけるデータ型を指定します。下記参照。
r_type:	応答 (msgdata) 内のデータ型を指定します。下記参照。

数値入力は以下の形式でできます。

10進数 (例 : 50)	(接頭文字なし)
8進数 (例 : 043)	接頭文字0 (ゼロ)
16進数 (例 : 0x1f)	接頭文字0x

- コマンドデータ型(c_type)

配列対応のデータ型では要素数を接尾文字[n] (nが要素数を指定) を使って指定できます。各データ要素は空白で区切る必要があります。

タイプ	配列対応	データ形式 (msgdataで提供)
BOOL	Yes	1
SINT8	Yes	-25
SINT16	Yes	2345
SINT32	Yes	-2569
UINT8	Yes	245
UINT16	Yes	40000
UINT32	Yes	32
CHAR	Yes	A
STRING	No	「abcde」 注: バックスラッシュ (「\」) を前に付けると、文字列に引用符を含めることができます。 例: 「通常これを「\Egg\」と呼ぶ
FLOAT	はい	5.6538e2
NONE	No	コマンドはデータを保持できないので、データ型もありません

- 応答データ型(r_type)

配列対応のデータ型では要素数を接尾文字[n] (n が要素数を指定) を使って指定できます。

タイプ	配列対応	データ形式 (msgdataで提供)
BOOL	あり	オプションとして、値 (trueまたはfalse) に基づくメッセージとBOOLデータの交換が可能です。こうした場合には関数から返される実際のデータ型はSTRINGになります。 構文: BOOL<true><false> 配列ならこのフォーマットは BOOL[n]<true><false>になります。
SINT8	あり	-
SINT16	あり	-
SINT32	あり	-
UINT8	あり	この型は ENUM データ型をオブジェクトから読み出すときにも使用できます。このような場合、実際のENUM値が返されます。
UINT16	あり	-
UINT32	あり	-
CHAR	あり	-
STRING	不可	-
ENUM	不可	このデータ型を使用する時、ABCCMessage()関数は最初にENUM値を読み取りません。その次に'Get Enum String'-コマンドを発行し、実際の列挙文字列を取得します。応答の実際のデータ型はSTRINGになります。
FLOAT	あり	-
NONE	不可	応答はデータを保持しないため、データ型はありません



メッセージが対象オブジェクトへ透過的に渡されることに注意することは重要です。SSIエンジンはオブジェクトアドレッシングスキーム違反をチェックしないため、メッセージデータを (不正に) 含んだ不正のGet_Attribute要求などが、明らかに不正であるにもかかわらず、無修正のままオブジェクトに渡されます。注意を怠ると、データ損失やその他の望ましくない影響が発生する場合があります。

例 (Get_Attribute):

この例はIPアドレスをprintf()と ABCCMessage()で取得する方法を示します。

```
<?--#exec cmd_argument='printf( "%u.%u.%u.%u",
ABCCMessage(4,3,1,5,0,0,NONE,UINT8[4] ) )'-->
```

Variable (変数)	値	コメント
object	4	ネットワークコンフィグレーションオブジェクト (04h)
instance	3	Instance #3 (IPアドレス)
command	1	Get_attribute
ce0	5	アトリビュート #5
ce1	0	-
msgdata	0	-
c_type	NONE	コマンドメッセージはデータを保持しない
r_type	UINT8[4]	4個の符号なし8ビット整数

例 (Set_Attribute):

この例は、scanf() とABCCMessage() を使用してIPアドレスを設定する方法を示します。scanf() によって解析済みのフォームデータを使用するようにモジュールに指示する特別なパラメータ「ARG」に注意してください。

```
<?--#exec cmd_argument='scanf("IP", "%u.%u.%u.%u",
ABCCMessage(4,3,2,5,0,ARG,UINT8[4],NONE ) )'-->
```

Variable (変数)	値	コメント
object	4	ネットワークコンフィグレーションオブジェクト (04h)
instance	3	Instance #3 (IPアドレス)
command	2	Set_attribute
ce0	5	アトリビュート #5
ce1	0	-
msgdata	ARG	scanf() 呼び出しによって解析されたデータを使用する
c_type	UINT8[4]	4個の符号なし8ビット整数の配列
r_type	NONE	応答メッセージはデータを保持しない

エラー

オブジェクト要求がエラーになった場合、応答内のエラーコードが評価され、以下のように読みやすい形式に変換されます。

エラーコード	出力
0	「Unknown error」
1	「Unknown error」
2	「Invalid message format」
3	「Unsupported object」
4	「Unsupported instance」
5	「Unsupported command」
6	「Invalid CmdExt[0]」
7	「Invalid CmdExt[1]」
8	「Attribute access is not set-able」
9	「Attribute access is not get-able」
10	「Too much data in msg data field」
11	「Not enough data in msg data field」
12	「Out of range」
13	「Invalid state」
14	「Out of resources」
15	「Segmentation failure」
16	「Segmentation buffer overflow」
17~255	「Unknown error」

下記も参照してください。

[SSI出力の設定, ページ 55](#)

8.5 SSI出力の設定

オプションとして、SSI出力はファイル \output.cfgを追加することにより永続的に変更されます。

ファイルフォーマット:

[ABCCMessage_X] 0: 「Success string」 1: 「Error string 1」 2: 「Error string 2」 ... 16 「Error string 16」	各エラーコードは1～16のラベルが付いた専用出力文字列に対応します。 エラー, ページ 54 を参照
[GetConfigItem_X] 0: 「Success string」 1: 「Authentication error string」 2: 「File open error string」 3: 「Tag not found string」	ファイル名を含めるには「%s」を使用してください。
[SetConfigItem_X] 0: 「Success string」 1: 「Authentication error string」 2: 「File open error string」 3: 「File write error string」	ファイル名を含めるには「%s」を使用してください。
[IncludeFile_X] 0: 「Success string」 1: 「Authentication error string」 2: 「File read error string」	ファイル名を含めるには「%s」を使用してください。
[scanf_X] 0: 「Success string」 1: 「Parsing error string」	-
[ChangeLanguage_X] 0: 「Success string」 1: 「Change error string」	-

上記のすべてのコンテンツは、各タグの値「X」を言語別に変更して、ファイルに複数回含めることができます。これにより、モジュールは言語設定に基づいて正しい出力文字列を選択します。選択された言語に関して情報が見つからない場合、デフォルトのSSI出力を使用します。

Xの値	Language
0	英語
1	ドイツ語
2	スペイン語
3	イタリア語
4	フランス語

下記も参照してください。

-

[SsiOutput\(\), ページ 41](#)

9 JSON

9.1 基本情報

JSONは、JavaScript Object Notationの頭字語で、体系的で直感的な方法でデータを保存および交換するためのオープンスタンダード形式です。Anybus CompactComでは、Anybus CompactComのWebサーバとWebアプリケーション間で名前と値のペアからなるデータオブジェクトを送信するために使用されます。オブジェクトメンバーには決まった順序がないため、値のペアはランダムに現れます。JavaScriptsはダイナミック Web ページを作成して値を表すために使用されます。オプションとして、コールバックをJSONP出力用のGET要求へ渡すこともできます。

JSONのほうが、読み出し・書き込みだけでなくWebページのサイズと見た目を動的に変えることもできるため、SSIより融通が利きます。Webページの作成方法の一例がこの章の最後にありますのでご参照ください。

9.1.1 エンコーディング

JSON 要求はUTF-8エンコードによります。モジュールはJSONの要求をUTF-8エンコードとして要求しますが、その他すべてのHTTP要求はISO-8859-1エンコードとして解釈されます。モジュールから送られるJSONのすべての応答はUTF-8でエンコードされ、Webから送られるその他のファイルはファイルシステムに保存されたときにエンコードされます。

9.1.2 アクセス

JSONリソースはパスワードで保護することをお勧めします。ルートディレクトリにweb_accs.cfgというファイルを追加して、パスワード保護を追加します (すべてのWebコンテンツが保護されます)。このファイルについては、本ドキュメントの「Webサーバ」セクションで説明しています。

9.1.3 エラー応答

モジュールがリクエストの解析または処理に失敗した場合、レスポンスにはAnybusエラーコードを持つエラーオブジェクトが含まれます。

```
{ "error" : 02 }
```

Anybusエラーコードは、『Anybus CompactCom 40 Software Design Guide』に一覧が掲載されています。

9.2 JSONオブジェクト

9.2.1 ADI

info.json

```
GET adi/info.json[?callback=<function>]
```

このオブジェクトは、ADI JSONインターフェースに関する情報を保持します。このデータはランタイム中は静的です。

Name	データ型	注
dataformat	番号	0 = リトルエンディアン 1 = ビッグエンディアン (値、最小、最大表記に影響する)
numadis	番号	ADIの合計数
webversion	番号	Web/JSON APIバージョン

JSONの応答例：

```
{ "dataformat": 0, "numadis": 123, "webversion": 1 }
```

data.json

```
GET adi/data.json?offset=<offset>&count=<count>[&callback=<function>]
GET adi/data.json?inst=<instance>&count=<count>[&callback=<function>]
```

これらのオブジェクト呼び出しは、<offset>または<instance>から始まる<count>個までのADI値のソート済みリストを取り出します。返される値は、ランタイム中に随時変化します。

要求データ：

Name	データ型	説明
offset	番号	オフセットは、最初に要求されたADIの「オーダー番号」です。 最初に実装されたADIが常にオーダー番号0を取得します。既存のADI値の<count>番号が返されます。すなわち、存在しないADIはスキップされます。
inst	番号	最初に要求されたADIのインスタンス番号。 ADI値の<count>番号が返されます。存在しないADIについてはNull値が返されます
count	文字列	要求されたADI値の数
callback	番号	オプション。 JSON出力のコールバック機能。

応答データ：

Name	データ型	説明
—	文字列の配列	ADI値アトリビュートの文字列表現のソート済みリスト

JSONの応答例 (offsetが使用される場合)：

```
[ "FF", "A201", "01FAC105" ]
```

JSONの応答例 (instが使用される場合)：

```
[ "FF", "A201", null, null, "01FAC105" ]
```

metadata.json

```
GET adi/metadata.json?offset=<offset>&count=<count>[&callback=<function>]
GET adi/metadata.json?inst=<instance>&count=<count>[&callback=<function>]
```

これらのオブジェクト呼び出しは、<offset>または<instance>から開始して、<count>個までのADIのメタデータオブジェクトのソートされたリストを取得します。

返される情報は、ホストアプリケーションデータオブジェクト (FEh) で使用可能なアトリビュートの透過的な表現となります。各アトリビュートの内容について詳しくは、『Anybus CompactCom 40 Software Design Guide』を参照してください。

ADIメタデータはランタイム中は静的です。

要求データ :

Name	データ型	説明
offset	番号	オフセットは、最初に要求されたADIの「オーダー番号」です。 最初に実装されたADIが常にオーダー番号0を取得します。既存のADIの<count>番号に対応するメタデータオブジェクトが返されます。すなわち、存在しないADIはスキップされます。
inst	番号	最初に要求されたADIのインスタンス番号。 ADI値の<count>番号に対応するメタデータオブジェクトが返されます。存在しないADIについてはNullオブジェクトが返されます
count	文字列	要求されたADI値の数
callback	番号	オプション。 JSON出力のコールバック機能。

応答データ :

Name	データ型	説明
instance	番号	-
name	文字列	名前がなければ NULL。
numelements	番号	-
datatype	番号	-
min	文字列	Hexフォーマットの文字列に関する詳細は、 Hexフォーマットの説明, ページ 74 を参照してください。 最小値が無ければNULLであって構いません。
max	文字列	Hexフォーマットの文字列に関する詳細は、 Hexフォーマットの説明, ページ 74 を参照してください。 最大値が無いときはNULLであって構いません。
access	番号	ビット0: 読み出しアクセス ビット1: 書き込みアクセス

JSONの応答例 (offsetが使用される場合):

```
[
{
  "instance": 1,
  "name": "Temperature threshold",
  "numelements": 1,
  "datatype": 0,
  "min": "00",
  "max": "FF",
  "access": 0x03
},
{
  ...
}
]
```

JSONの応答例 (instが使用される場合):

```
[
{
  "instance": 1,
  "name": "Temperature threshold",
  "numelements": 1,
  "datatype": 0,
  "min": "00",
  "max": "FF",
  "access": 0x03
},
null,
null
{
  ...
}
]
```

metadata2.json

```
GET adi/metadata2.json?offset=<offset>&count=<count> [&callback=<function>]
GET adi/metadata2.json?inst=<instance>&count=<count> [&callback=<function>]
```

これはADIに関する完全な情報を提供するメタデータ機能の拡張版です。この拡張バージョンは、構造体などのより複雑なデータ型を記述するために必要です。

提供される情報はホストアプリケーションデータオブジェクト (FEh) で利用可能なアトリビュートの透過的な表現です。各アトリビュートの内容について詳しくは、『Anybus CompactCom 40 Software Design Guide』を参照してください。

ADIメタデータはランタイム中は静的です。

要求データ：

Name	データ型	説明
offset	番号	オフセットは、最初に要求されたADIの「オーダー番号」です。最初に実装されたADIが常にオーダー番号0を取得します。既存のADIの<count>番号に対応するメタデータオブジェクトが返されます。すなわち、存在しないADIはスキップされます。
inst	番号	最初に要求されたADIのインスタンス番号。ADI値の<count>番号に対応するメタデータオブジェクトが返されます。存在しないADIについてはNullオブジェクトが返されます
count	文字列	要求されたADI値の数
callback	番号	オプション。 JSONP出力のコールバック機能。

応答データ：

Name	データ型	説明
instance	番号	-
numelements	数字配列	-
datatype	数字配列	データ型の配列。 構造体と変数の場合、各配列要素はインスタンス値の対応する要素のデータ型を定義します。配列の場合、1つの配列要素がインスタンス値のすべての要素のデータ型を定義します。
descriptor		記述子の配列。 構造体と変数の場合、各配列要素はインスタンス値の対応する要素の記述子を定義します。配列の場合、1つの配列要素がインスタンス値のすべての要素の記述子を定義します。
name		名前がなければNULL。
min	文字列	Hexフォーマットの文字列に関する詳細は、 Hexフォーマットの説明, ページ 74 を参照してください。 最小値が無ければNULLであって構いません。
max	文字列	Hexフォーマットの文字列に関する詳細は、 Hexフォーマットの説明, ページ 74 を参照してください。 最大値が無いときはNULLであって構いません。
default	文字列	Hexフォーマットの文字列に関する詳細は、 Hexフォーマットの説明, ページ 74 を参照してください。 デフォルト値が無いときはNULLにできます。
numsubelements	数字配列	構造体と変数の場合、各配列要素はインスタンス値の対応する要素の下部要素の数を定義します。 存在しない場合はNULLにできます。
elementname	文字列の配列	名前の配列。各インスタンス値要素に1つずつ。 存在しない場合はNULLにできます。

JSONの応答例 (offsetが使用される場合):

```
[
{
  "instance": 1,
  "numelements": 1,
  "datatype": [0 ],
  "descriptor": [9 ],
  "name": "Temperature threshold",
  "max": "FF",
  "min": "00",
  "default": "00",
  "numsubelements": null
  "elementname": null },
{
  ...
}
]
```

JSONの応答例 (インスタンス):

```
[
{
  "instance": 1,
  "numelements": 1,
  "datatype": [0 ],
  "descriptor": [9 ],
  "name": "Temperature threshold",
  "max": "FF",
  "min": "00",
  "default": "00",
  "numsubelements": null
  "elementname": null
},
null,
null
{
  ...
}
]
```

enum.json

```
GET adi/enum.json?inst=<instance> [&value=<element>] [&callback=<function>]
```

このオブジェクト呼び出しは特定のインスタンスの列挙文字列のリストを取得します。

ADI列挙文字列はランタイム中は静的です。

要求データ :

Name	データ型	説明
inst	番号	列挙文字列を取得するためのADIのインスタンス番号。
value	番号	オプション。指定した場合、要求された<value>の列挙文字列のみが返されます。
callback	文字列	オプション。JSONP出力のコールバック機能。

応答データ :

Name	データ型	説明
string	文字列	対応する値の文字列表現。
value	番号	文字列表現に対応する値。

JSONの応答例 :

```
[
  {
    "string": "String for value 1",
    "value": 1
  },
  {
    "string": "String for value 2",
    "value": 2
  },
  {
    ...
  }
]
```


update.json

```
POST adi/update.json
```

フォームデータ :

```
inst=<instance>&value=<data> [&elem=<element>] [&callback=<function>]
```

ADIの値アトリビュートを更新します。

要求データ :

Name	データ型	説明
inst	番号	ADIのインスタンス番号
value	文字列	設定する値。 値アトリビュートが数字である場合は、Hexフォーマットである必要があります。 詳細は Hexフォーマットの説明, ページ 74 を参照してください。
elem	番号	オプション。 指定した場合、ADI値の単一要素のみが設定されます。これにより、<data>には指定された<element> の値のみが含まれます。
callback	文字列	オプション。 JSONP出力のコールバック機能。

応答データ :

Name	データ型	注
result	番号	0 = success Anybus CompactComエラーコードが使用されます。『Anybus CompactCom 40 Software Design Guide』を参照してください。

```
{ "result" : 0 }
```

9.2.2 モジュール

info.json

```
GET module/info.json
```

応答データ :

Name	データ型	説明
modulename	文字列	-
serial	文字列	32ビットのHex ASCII
fwver	数字配列	(メジャー、マイナー、ビルド)
uptime	数字配列	uptime(稼働時間)は、2つの32ビット値の配列で[High、Low] msとして実装されま す。
cpuload	番号	CPU負荷率%
fwvertext	文字列	文字表記によるファームウェアバージョン
vendorname	文字列	ベンダー名 (アプリケーションオブジェクト(FFh)、インスタンスアトリビュート #8)
hwvertext	文字列	文字表記によるハードウェアバージョン
networktype	番号	ネットワークのタイプ(ネットワークオブジェクト(03h)、インスタンスアトリ ビュート#1)

JSONの応答例 :

```
{
  "modulename": "ABCC M40",
  "serial": "ABCDEF00",
  "fwver": [ 1, 5, 0 ],
  "uptime": [ 5, 123456 ],
  "cpuload": 55,
  "fwvertext": "1.05.02",
  "vendorname": "HMS Industrial Networks",
  "hwvertext": "2",
  "networktype": 133,
}
```

9.2.3 ネットワーク

ethstatus.json

```
GET network/ethstatus.json.
```

Name	データ型	説明
mac	文字列	6 バイトの16進数
comm1	オブジェクト	下表のオブジェクト定義参照
comm2	オブジェクト	下表のオブジェクト定義参照

Commオブジェクトの定義:

Name	データ型	説明
link	番号	0: リンクなし 1: リンクあり
speed	番号	0: 10 Mbit 1: 100 Mbit
duplex	番号	0: 半二重 1: 全二重

JSONの応答例 :

```
{
  "mac": "003011FF0201",
  "comm1": {
    "link": 1,
    "speed": 1,
    "duplex": 1
  },
  "comm2": {
    "link": 1,
    "speed": 1,
    "duplex": 1
  }
}
```

ipstatus.json & ipconf.json

これら二つのオブジェクトのデータフォーマットは同じです。オブジェクトipconf.jsonは設定済みのIP構成を返し、ipstatus.jsonは現在使用中の実際の値を返します。ipconf.jsonではIP構成の変更もできます。

```
GET network/ipstatus.json
```

または

```
GET network/ipconf.json
```

Name	データ型	注
dhcp	番号	-
addr	文字列	-
subnet	文字列	-
gateway	文字列	-
dns1	文字列	-
dns2	文字列	-
hostname	文字列	-
domainname	文字列	-

```
{
  "dhcp":0,
  "addr":"192.168.0.55",
  "subnet":"255.255.255.0",
  "gateway":"192.168.0.1",
  "dns1":"10.10.55.1",
  "dns2":"10.10.55.2",
  "hostname":"abcc123",
  "domainname":"hms.se"
}
```

IP設定を変更するには、network/ipconf.jsonを使用します。これは上記リストから任意の数の引数を受け入れます。値は同じ形式でなければなりません。

例:

```
GET ipconf.json?dhcp=0&addr=10.11.32.2&hostname=abcc123&domainname=hms.se
```

ethconf.json

```
GET network/ethconf.json
```

Name	データ型	注
mac	文字列	-
comm1	番号	-
comm2	番号	モジュール内で2つのイーサネットポートが有効になっている場合にのみ存在します。

「comm1」および「comm2」の値は、ネットワークコンフィグレーションオブジェクト、インスタンス#7および#8から読み出されます。

```
{
  "mac": [00, 30, 11, FF, 02, 01],
  "comm1": 0,
  "comm2": 4
}
```

パラメータ「comm1」および「comm2」は、次のようにGET要求に引数として追加することで設定可能です。

```
GET network/ethconf.json?comm1=0&comm2=4
```

モジュールによる要求処理が失敗に終わった場合、パラメータ「comm1」および「comm2」は、以下のようなAnybusエラーコードを含むエラーオブジェクトを保持していることがあります。

```
{
  "mac": [00, 30, 11, FF, 02, 01],
  "comm1": 0,
  "comm2": { error: 14 },
}
```

Anybus CompactComエラーコードが使用されます。『Anybus CompactCom 40 Software Design Guide』を参照してください。

ifcounters.json

```
GET network/ifcounters.json?port=<port>
```

引数<port>の有効値は0、1、2です。

- 引数<port>の有効値は0、1、2です。
- ポート番号0というオプションは、内部ポート (CPU ポート) を表します。
- ポート番号2というオプションは、モジュール内で2つのイーサネットポートがアクティブになっている場合にのみ有効です。

Name	データ型	説明
inoctets	番号	IN: バイト
inucast	番号	IN: ユニキャストパケット数
innucast	番号	IN: ブロードキャストとマルチキャストパケット数
indiscards	番号	IN: 破棄パケット数
inerrors	番号	IN: エラー
inunknown	番号	IN: 未対応のプロトコルタイプ
outoctets	番号	OUT: バイト
outucast	番号	OUT: ユニキャストパケット数
outnucast	番号	OUT: ブロードキャストとマルチキャストパケット数
outdiscards	番号	OUT: 破棄パケット数
outerrors	番号	OUT: エラー数

mediacounters.json

```
GET network/mediacounters.json?port=<port>
```

引数<port>は1か2しか取りません。

ポート番号2というオプションは、モジュール内で2つのイーサネットポートがアクティブになっている場合にのみ有効です。

Name	データ型	説明
align	番号	長さが8進数整数以外の受信フレーム数
fcs	番号	FCSチェックに失格した受信フレーム数
singlecoll	番号	一回だけコリジョンが発生して転送に成功したフレーム数
multicoll	番号	二回以上コリジョンが発生して転送に成功したフレーム数
latecoll	番号	パケット伝送に対して512ビット回より後でコリジョンが検出された回数
excesscoll	番号	コリジョンが多すぎて伝送に失敗したフレーム数
sqetest	番号	SQEテストエラーが生成された回数
deferredtrans	番号	第一回の伝送試行が媒体ビジーにより遅延しているフレーム数
macrecerr	番号	内部MACのサブレイヤー受信エラーにより受信が失敗したフレーム数
mactranserr	番号	内部MACのサブレイヤー送信エラーにより伝送が失敗したフレーム数
cserr	番号	搬送波検出条件が損失したまたはフレーム転送の試行時にアサートされなかった回数
toolong	番号	最大許容フレームサイズを超過した受信フレーム数

nwstats.json

```
GET network/nwstats.json
```

このオブジェクトは利用可能な統計データを一覧表示します。利用可能なデータは製品によって異なります。

出力例:

```
[
or
[ { "identifier": "eipstats", "title": "EtherNet/IP Statistics" } ]
or
[ { "identifier": "eitstats", "title": "Modbus TCP Statistics" } ]
or
[
{ "identifier": "bacnetipstats",
"title": "BACnet/IP Statistics" },
{ "identifier": "bacnetaplserverstats",
"title": "BACnet Application Layer Server Statistics" },
{ "identifier": "bacnetaplclientstats",
"title": "BACnet Application Layer Client Statistics" }
{ "identifier": "bacnetalarmstats",
"title": "BACnet Alarm and Event Module Statistics" }
]
or
[
{ "identifier": "eplifcounters",
"title": "IT Interface Counters" }
]
or
[
{ "identifier": "ectstats",
"title": "EtherCAT Statistics" },
{ "identifier": "eoeifcounters",
"title": "EoE Interface Counters" }, ]
or
[ { "identifier" : "pnpof", "title" : "Fiber Optical Statistics" } ]
```

ネットワーク固有の統計を取得 (<ID>は以前のコマンドから返される“identifier”値):

```
GET network/nwstats.json?get=<ID>
```

「eipstats」

```
[
{ "name": "Established Class1 Connections", "value": 0 },
{ "name": "Established Class3 Connections", "value": 1 }
{ "name": "Connection Open Request", "value": 0 },
{ "name": "Connection Open Format Rejects", "value": 0 },
{ "name": "Connection Open Resource Rejects", "value": 0 },
{ "name": "Connection Open Other Rejects", "value": 0 },
{ "name": "Connection Close Requests", "value": 0 },
{ "name": "Connection Close Format Rejects", "value": 0 },
{ "name": "Connection Other Rejects", "value": 0 },
{ "name": "Connection Timeouts", "value": 0 },
]
```


「eitstats」

```
[
{ "name": "Modbus Connections", "value": 0 },
{ "name": "Connection ACKs", "value": 1 }
{ "name": "Connection NACKs", "value": 0 },
{ "name": "Connection Timeouts", "value": 0 },
{ "name": "Process Active Timeouts", "value": 0 },
{ "name": "Processed messages", "value": 0 },
{ "name": "Incorrect messages", "value": 0 },
]
```

「bacnetipstats」

```
[
{ "name": "Unconfirmed server requests received", "value": 0 },
{ "name": "Unconfirmed server requests sent", "value": 1 }
{ "name": "Unconfirmed client requests sent", "value": 0 },
]
```

「bacnetapserverstats」

```
[
{ "name": "Active transactions", "value": 0 },
{ "name": "Max Active transactions", "value": 1 }
{ "name": "Tx segments sent", "value": 0 },
{ "name": "Tx segment ACKs received", "value": 0 },
{ "name": "Tx segment NAKs received", "value": 0 },
{ "name": "Rx segments received", "value": 0 },
{ "name": "Rx segment ACKs sent", "value": 0 },
{ "name": "Duplicate Rx segment ACKs sent", "value": 0 },
{ "name": "Rx segment NAKs sent", "value": 0 },
{ "name": "Confirmed transactions sent", "value": 0 },
{ "name": "Confirmed transactions received", "value": 0 },
{ "name": "Tx segment timeouts", "value": 0 },
{ "name": "Rx segment timeouts", "value": 0 },
{ "name": "Implicit deletes", "value": 0 },
{ "name": "Tx timeout deletes", "value": 0 },
{ "name": "Rx timeout deletes", "value": 0 },
{ "name": "Tx aborts received", "value": 0 },
{ "name": "Rx aborts received", "value": 0 },
{ "name": "Transaction aborts sent", "value": 0 },
{ "name": "Transaction rejects sent", "value": 0 },
{ "name": "Transaction errors sent", "value": 0 },
]
```

「bacnetapclientstats」

```
[
{ "name": "Active transactions", "value": 0 },
{ "name": "Max Active transactions", "value": 1 }
{ "name": "Tx segments sent", "value": 0 },
{ "name": "Tx segment ACKs received", "value": 0 },
{ "name": "Tx segment NAKs received", "value": 0 },
{ "name": "Rx segments received", "value": 0 },
{ "name": "Rx segment ACKs sent", "value": 0 },
{ "name": "Duplicate Rx segment ACKs sent", "value": 0 },
{ "name": "Rx segment NAKs sent", "value": 0 },
{ "name": "Confirmed transactions sent", "value": 0 },
{ "name": "Confirmed transactions received", "value": 0 },
{ "name": "Tx segment timeouts", "value": 0 },
{ "name": "Rx segment timeouts", "value": 0 },
{ "name": "Implicit deletes", "value": 0 },
{ "name": "Tx timeout deletes", "value": 0 },
{ "name": "Rx timeout deletes", "value": 0 },
{ "name": "Tx aborts received", "value": 0 },
{ "name": "Rx aborts received", "value": 0 },
{ "name": "Transaction aborts sent", "value": 0 },
{ "name": "Transaction rejects sent", "value": 0 },
{ "name": "Transaction errors sent", "value": 0 },
]
```

「bacnetalarmstats」

```
[
{ "name": "COV Active subscriptions", "value": 0 },
{ "name": "COV Max active subscriptions", "value": 1 }
{ "name": "COV Lifetime subscriptions", "value": 0 },
{ "name": "COV Confirmed resumes", "value": 0 },
{ "name": "COV Unconfirmed resumes", "value": 0 },
{ "name": "COV Confirmed notifications sent", "value": 0 },
{ "name": "COV Unconfirmed notifications sent", "value": 0 },
{ "name": "COV Confirmed notification errors", "value": 0 },
{ "name": "AE Active events", "value": 0 },
{ "name": "AE Active NC recipients", "value": 0 },
{ "name": "AE Confirmed resumes", "value": 0 },
{ "name": "AE UnConfirmed resumes", "value": 0 },
{ "name": "AE Confirmed notifications sent", "value": 0 },
{ "name": "AE UnConfirmed notifications sent", "value": 0 },
{ "name": "AE Confirmed notification errors", "value": 0 },
{ "name": "AE DAB lookup errors", "value": 0 },
]
```

「replifcounters」

```
[
{ "name": "In Octets", "value": 22967 },
{ "name": "In Ucast Packets", "value": 121 },
{ "name": "In NUcast Packets", "value": 31 },
{ "name": "In Discards", "value": 0 },
{ "name": "In Errors", "value": 0 },
{ "name": "In Unknown Protos", "value": 0 },
{ "name": "Out Octets", "value": 169323 },
{ "name": "Out Ucast Packets", "value": 168 },
{ "name": "Out NUcast Packets", "value": 16 },
{ "name": "Out Discards", "value": 0 },
{ "name": "Out Errors", "value": 0 },
]
```

「rectstats」

```
[
{ "name": "Logical EoE port link", "value": "Yes" },
{ "name": "Invalid frame counter IN port", "value": 1 },
{ "name": "Rx error counter IN port", "value": 1 },
{ "name": "Forwarded error counter IN port", "value": 1 },
{ "name": "Lost link counter IN port", "value": 1 },
{ "name": "Invalid frame counter OUT port", "value": 1 },
{ "name": "Rx error counter OUT port", "value": 1 },
{ "name": "Forwarded error counter OUT port", "value": 1 },
{ "name": "Lost link counter OUT port", "value": 1 },
]
```

「eoeifcounters」

```
[
{ "name": "In Octets", "value": 22967 },
{ "name": "In Ucast Packets", "value": 121 },
{ "name": "In NUcast Packets", "value": 31 },
{ "name": "In Discards", "value": 0 },
{ "name": "In Errors", "value": 0 },
{ "name": "In Unknown Protos", "value": 0 },
{ "name": "Out Octets", "value": 169323 },
{ "name": "Out Ucast Packets", "value": 168 },
{ "name": "Out NUcast Packets", "value": 16 },
{ "name": "Out Discards", "value": 0 },
{ "name": "Out Errors", "value": 0 },
]
```


「pnpof」

```
[
{ "name" : "Port 1 Temperature (C)", "value" : "41.37" },
{ "name" : "Port 1 Power Budget (dB)", "value" : "23.0" },
{ "name" : "Port 1 Power Budget Status", "value" : "OK" },
{ "name" : "Port 2 Temperature (C)", "value" : "40.57" },
{ "name" : "Port 2 Power Budget (dB)", "value" : "0.0" },
{ "name" : "Port 2 Power Budget Status", "value" : "OK" }
]
```

9.2.4 サービス

smtp.json

```
GET services/smtp.json
```

 設定を取得する際にパスワードは返されません。

Name	データ型	注
server	文字列	IPアドレスまたはメールサーバの名前、「mail.hms.se」など
user	文字列	-

```
[
  { "server": "192.168.0.55"},
  { "user": "test"}
]
```

設定：

フォームデータ：

```
[
  [server=192.168.0.56]&[user=test2]&[password=secret],
]
```

9.2.5 Hexフォーマットの説明

メタデータの最大、最小、デフォルトフィールドおよびADI値は、ASCII 16進表現でエンコードされたバイナリデータです。データ型が整数の場合は、使用されるエンディアン種別はadi/info.jsonで見つかったデータフォーマットフィールドによって決まります。

例:

値5がUINT16とエンコードされ、データフォーマット=0(リトルエンディアン)の場合:

```
0500
```

文字配列「ABC」がCHAR[3]としてエンコードされた場合 (CHARにはデータフォーマットは関係ありません):

```
414243
```

9.3 例

この例は、モジュールからモジュール名と CPU 負荷率をフェッチして、Web に表すウェブページの作成方法を示します。このコードを含むファイルは内蔵ファイルシステムに保存しなければならず、結果は共通ブラウザに表示されます。

```
<html>
<head>
<title>Anybus CompactCom</title>

<!-- Imported libs -->
<script type="text/javascript" src="vfs/js/jquery-1.9.1.js"></script>
<script type="text/javascript" src="vfs/js/tmpl.js"></script>
</head>
<body>
<div id="info-content"></div>
<script type="text/x-tmpl" id="tmpl-info">
<b>From info.json</b><br>
Module name:
{%=o.modulename%}<br>

CPU Load:
{%=o.cpubload%}<br>
</script>
<script type="text/javascript">
$.getJSON( "/module/info.json", null, function(data){
$("#info-content").html( tmpl("tmpl-info", data ) );
});
</script>
</body>
</html>
```

10 CIPオブジェクト

10.1 基本情報

この章では、本モジュールで実装されているCIPオブジェクトについて説明します。これらのオブジェクトはネットワークからアクセスできますが、ホストアプリケーションからは直接アクセスできません。

必須オブジェクト

- [Identity オブジェクト \(01h\)](#), ページ 78
- [メッセージルータ \(02h\)](#), ページ 81
- [アセンブリオブジェクト \(04h\)](#), ページ 82
- [コネクションマネージャ \(06h\)](#), ページ 85
- [QoS オブジェクト \(48h\)](#), ページ 97
- [TCP/IPインターフェースオブジェクト \(F5h\)](#), ページ 106
- [Ethernet リンクオブジェクト \(F6h\)](#), ページ 109

CIP Energy オブジェクト:

- [Base Energy オブジェクト \(4Eh\)](#), ページ 98
- [Power Management オブジェクト \(53h\)](#), ページ 100

オプションオブジェクト:

- [パラメータオブジェクト \(0Fh\)](#), ページ 89
- [Time Syncオブジェクト \(43h\)](#), ページ 92
- [DLRオブジェクト \(47h\)](#), ページ 96
- [ポートオブジェクト\(F4h\)](#), ページ 104 (オプション)

ベンダー固有のオブジェクト:

- [ADIオブジェクト \(A2h\)](#), ページ 102

CIP転送機能を使用して追加のCIPオブジェクトをホストアプリケーションに実装することができます。[EtherNet/IPホストオブジェクト \(F8h\)](#), ページ 175ならびにProcess_CIP_Object_Requestコマンドの詳細情報を参照してください。

非コネクション型CIPルーティングに対応します。即ち、コネクションセットアップをせずにメッセージをデバイスに送信できます。

10.2 ステータスコードの変換

オブジェクトがアプリケーションから要求されているときにエラーが発生するとエラーコードが返されます。これらのAnybus CompactComエラーコードは下表に基づいてCIPステータスコードに変換されます。

Anybus CompactCom 40エラーコード		CIPステータスコード	
値	エラー	値	状態
00h	Reserved	1Eh	埋め込みサービスエラー
01h	Reserved	1Eh	埋め込みサービスエラー
02h	メッセージフォーマットが不正	1Eh	埋め込みサービスエラー
03h	サポートしていないオブジェクト	05h	バスのターゲットが不明
04h	サポートしていないインスタンス	05h	バスのターゲットが不明
05h	サポートしていないコマンド	08h	サービス未サポート
06h	不正なCmdExt(0)	14h	「サポートされていないアトリビュート」など、この応答を返すAnybus CompactCom サービスに依存
07h	不正なCmdExt(1)	-	この応答を返すAnybus CompactComサービスに依存
08h	アトリビュートはset出来ない	0Eh	アトリビュートはset出来ない
09h	アトリビュートはget出来ない	2Ch	アトリビュートはget出来ない
0Ah	データが多すぎる	15h	データが多すぎる
0Bh	データが足りない	13h	データが足りない
0Ch	範囲外	09h	無効なアトリビュート値
0Dh	無効な状態	0Ch	オブジェクト状態の不一致
0Eh	リソースが足りない	02h	リソース利用不可
0Fh	分割エラー	1Eh	埋め込みサービスエラー
10h	分割バッファオーバーフロー	23h	バッファオーバーフロー
11h	値が高すぎる	09h	無効なアトリビュート値
12h	値が低すぎる	09h	無効なアトリビュート値
13h	アトリビュートが制御された	0Fh	許可 / 権限チェックに失格
14h	メッセージチャンネルが小さすぎる	11h	応答データがサイズ超過
FFh	オブジェクト固有のエラー	1Fh	ベンダー固有のエラー。追加のエラーコードはEtherNet/IPでは送信されません
その他	-	1Eh	埋め込みサービスエラー

Anybus CompactComのエラーコードに関する詳細は、『Anybus CompactCom 40 Software Design Guide』を参照してください。

10.3 Identity オブジェクト (01h)

カテゴリ

拡張

オブジェクトの説明

Identity オブジェクトはモジュールの識別および基本情報を提供します。

オブジェクトは複数のインスタンスに対応します。唯一の必須インスタンスであるインスタンス1で製品全体を記述します。アプリケーションはこれでネットワークにあるノードを判断し、EDSファイルをネットワーク上の製品と照合します。その他 (オプション) のインスタンスは製品の異なる部分例えばソフトウェアについて記述しています。

モジュール式デバイス機能が有効なときは、スロットにあるモジュールの一覧を検索することが出来ます。そして、クラスアトリビュート100へget要求を送信する事でネットワークマスターが利用できるようにします。

インスタンスアトリビュート1~7はEtherNet/IP ホストオブジェクトを実装することによってカスタム化されます。

追加のアイデンティティインスタンスもCIPアイデンティティホストオブジェクト (ホストアプリケーションオブジェクト) を実装することによって登録できます。

下記も参照してください。

- [EtherNet/IPホストオブジェクト \(F8h\), ページ 175](#)
- [CIP Identity ホストオブジェクト \(EDh\), ページ 166](#)

サポートされているサービス

クラス :	Get_Attribute_Single
	Get_Attributes_All
インスタンス :	Get_Attribute_Single
	Set_Attribute_Single
	Get_Attributes_All
	Reset

クラスアトリビュート

#	名前	アクセス	タイプ	値
1	Revision	Get	UINT	0001h (オブジェクトのレビジョン)
2	Max instance	Get	UINT	最大インスタンス番号
3	Number of instances	Get	UINT	インスタンスの数
100	Module ID List	Get	UINT32 の配列	モジュラーデバイス 機能が有効な場合、このアトリビュートに対する要求はホストアプリケーション内のモジュラーデバイスオブジェクトに対してGet_List要求を生成します。

インスタンスアトリビュート

アトリビュート#1~4と#6~7はEtherNet/IPホストオブジェクトを実装することによってカスタム化されます ([EtherNet/IPホストオブジェクト \(F8h\)](#), ページ 175を参照)。

#	名前	アクセス	タイプ	値/説明
1	Vendor ID	Get	UINT	005Ah (HMS Industrial Networks)
2	Device Type	Get	UINT	002Bh (ジェネリックデバイス)
3	Product Code	Get	UINT	0037h (Anybus CompactCom 40 EtherNet/IP)
4	Revision	Get	構造体: USINT USINT	メジャーおよびマイナーなファームウェアリビジョン
5	Status	Get	WORD	下表のデバイスステータスを参照
6	Serial Number	Get	UDINT	一意のシリアル番号 (HMSで割当)
7	Product Name	Get	SHORT_STRING	「Anybus CompactCom 40 EtherNet/IP (TM)」
11	Active language	Set	構造体: USINT USINT USINT	このインスタンスに送られた要求はアプリケーションオブジェクトへ転送されます。要求が承認されると、モジュールはこれに伴い言語を更新します。
12	Supported Language List	Get	以下の配列: 構造体: USINT USINT USINT	ホストアプリケーションが対応している言語の一覧。このリストはアプリケーションオブジェクトから読み出され、CIP標準に変換されます。デフォルトでは唯一の対応言語は英語です。さらに他の言語も有効にするには、アプリケーションがアプリケーションオブジェクトに対応するアトリビュートを実装する必要があります。

デバイスステータス

bit	名前
0	Module Owned
1	(予約)
2	Configured(コンフィグ済み) このビットは製品が入荷時のもの以外にその他の設定があるのかについて表します。この値は、アプリケーションオブジェクトに設定されているまたはモジュールの NVストレージがデフォルトから変更された場合には、trueに設定されます。
3	(予約)
4...7	拡張デバイス ステータス: 値: 意味: 0000b 不明 0010b I/O接続障害 0011b I/O接続未確立 0100b 不揮発コンフィグレーション不正 0101b 重大な障害 0110b コネクションがRunモード 0111b コネクションがIdleモード (その他) (予約)
8	マイナーな復旧可能な障害が発生した場合にセットされる。詳細は次の章を参照 診断オブジェクト (02h), ページ 116
9	マイナーな復旧不可能な障害が発生した場合にセットされる。詳細は次の章を参照 診断オブジェクト (02h), ページ 116
10	重大な復旧可能な障害が発生した場合にセットされる。詳細は次の章を参照 診断オブジェクト (02h), ページ 116
11	重大な復旧不可能な障害が発生した場合にセットされる。詳細は次の章を参照 診断オブジェクト (02h), ページ 116
12~15	(予約)

サービス詳細: Reset



このサービスは、ファンクショナルセーフティオブジェクト (E8h)でセーフティが有効なときは対応しません。

モジュールはネットワークからホストアプリケーションへのリセット要求を転送します。ネットワークリセット処理に関する詳細については、全般的な『Anybus CompactCom 40 Software Design Guide』を参照してください。

EtherNet/IPに関するネットワークリセット要求には下記の二種類があります。どちらのタイプもアプリケーションによってサポートされている必要があります。

タイプ0: パワーサイクルリセット

このサービスはモジュールのパワーサイクルをエミュレートし、Anybusリセットタイプ0 (パワーサイクル) に対応します。詳細については、全般的な『Anybus CompactCom 40 Software Design Guide』を参照してください。

タイプ1: 工場出荷時リセット

このサービスは購入時の設定にリセットするもので、Anybusリセットタイプ2 (電源のオフオン + 出荷時設定) に相当します。詳細については、全般的な『Anybus CompactCom 40 Software Design Guide』を参照してください。

10.4 メッセージルータ (02h)

カテゴリ

拡張

オブジェクトの説明

メッセージルータオブジェクトは、クライアントが物理的モジュールに存在する任意のオブジェクトクラスまたはインスタンスへのサービスをアドレスすることができるメッセージ接続ポイントを提供します。

これはAnybus CompactComモジュールにおいてオブジェクト要求を送るために内部で使用されます。

サポートされているサービス

クラス : -

インスタンス : -

クラスアトリビュート

-

インスタンスアトリビュート

-

10.5 アセンブリオブジェクト (04h)

カテゴリ

拡張

オブジェクトの説明

アセンブリオブジェクトは静的アセンブリを使用し、ホストアプリケーションによって送信/受信されたプロセスデータを保持します。これはデータを各オブジェクトとの間で一本の接続によって送受信することを可能にします。デフォルトアセンブリインスタンスIDはベンダー固有の範囲になります。

アセンブリマッピングオブジェクトが実装されている場合、アプリケーションは最大で6個のconsuming インスタンスと6個のproducing インスタンスを生成してサポートすることができます。

「入力」および「出力」という用語はネットワークからの観点で定義されています。入力はデータをネットワーク上に生成し、出力はデータをネットワークから消費します。

下記も参照してください。

- [EtherNet/IPホストオブジェクト \(F8h\), ページ 175](#)
- アセンブリマッピングオブジェクト (『Anybus CompactCom 40 Software Design Guide』を参照)

サポートされているサービス

クラス : Get_Attribute_Single

インスタンス : Get_Attribute_Single

Set_Attribute_Single

クラスアトリビュート

#	名前	アクセス	タイプ	値
1	Revision	Get	UINT	0002h (オブジェクトリビジョン)
2	Max instance	Get	UINT	最大インスタンス番号

インスタンス03h アトリビュート (ハートビート、Input-Only)

このインスタンスはInput-Onlyコネクションのハートビートとして使用されます。Forward_Open要求におけるハートビートインスタンスのデータサイズは0バイトであるべきですが、その他の値でも構いません。

#	名前	アクセス	タイプ	値/説明
3	Data	Set	N/A	-(このアトリビュートのデータサイズはゼロ)
4	Size	Get	UINT	0 (アトリビュート 3 のバイト数)

このインスタンスのインスタンス番号は、対応するアトリビュートをEtherNet/IPホストオブジェクトに実装すると変更できます。

インスタンス04h アトリビュート (ハートビート、Listen-Only)

このインスタンスはListen-Onlyコネクションのハートビートとして使用されます。Forward_Open要求におけるハートビートインスタンスのデータサイズは0バイトであるべきですが、その他の値でも構いません。

#	名前	アクセス	タイプ	値/説明
3	Data	Set	N/A	-(このアトリビュートのデータサイズはゼロ)
4	Size	Get	UINT	0 (アトリビュート 3のバイト数)

このインスタンス用のインスタンス番号はEtherNet/IP ホストオブジェクトに対応するアトリビュートを実装すると変更できます。

インスタンス05h アトリビュート (コンフィグレーションデータ)

Forward_Openサービスによって送られるコンフィグレーションデータはこのインスタンスに書き込まれます。

#	名前	アクセス	タイプ	値/説明
3	Data	Set	N/A	-(Forward Openコマンドが設定データを含むときにアプリケーションに書き込まれたコンフィグレーションデータ) : (このアトリビュートのデータサイズはゼロ)
4	Size	Get	UINT	0 (アトリビュート 3のバイト数)

このインスタンス用のインスタンス番号はEtherNet/IP ホストオブジェクトに対応するアトリビュートを実装すると変更できます。

Set_Configuration_DataおよびGet_Configuration_Dataコマンドに関する詳細は、[EtherNet/IPホストオブジェクト \(F8h\)](#)、[ページ 175](#)を参照してください。

インスタンス06h アトリビュート (ハートビート、拡張 Input-Only)

このインスタンスは拡張 Input-Onlyコネクションのハートビートとして使用され、アトリビュートは含まれません。このインスタンスに対して行われたコネクションの状態はAnybus CompactComモジュールの状態には影響しません。つまり、コネクションがタイムアウトしても、モジュールはエラー状態になりません。Forward_Open要求におけるハートビートインスタンスのデータサイズは0バイトであるべきですが、その他の値でも構いません。

このインスタンスのインスタンス番号は、対応するアトリビュートをEtherNet/IPホストオブジェクトに実装すると変更できます。

#	名前	アクセス	タイプ	値/説明
3	Data	Set	N/A	-(このアトリビュートのデータサイズはゼロ)
4	Size	Get	UINT	0 (アトリビュート 3のバイト数)

インスタンス07h アトリビュート (ハートビート、拡張Listen-Only)

このインスタンスは拡張 Listen-Onlyコネクションとして使用され、アトリビュートは含まれません。このインスタンスを作るコネクションの状態はAnybus CompactCom 40 モジュールの状態に影響しません。即ち、コネクションがタイムアウトすると、モジュールはエラー状態になりません。Forward_Open要求におけるハートビートインスタンスのデータサイズは0バイトであるべきですが、その他の値でも構いません。

このインスタンスのインスタンス番号は、対応するアトリビュートをEtherNet/IPホストオブジェクトに実装すると変更できます。

#	名前	アクセス	タイプ	値/説明
3	Data	Set	N/A	-(このアトリビュートのデータサイズはゼロ)
4	Size	Get	UINT	0 (アトリビュート 3のバイト数)

インスタンス64hアトリビュート (Producing インスタンス)

このインスタンスのインスタンス番号は、対応するアトリビュートをEtherNet/IPホストオブジェクトに実装すると変更できます。

#	名前	アクセス	タイプ	値/説明
3	Produced Data	Get	BYTEの配列	このデータはライトプロセスデータに対応します。
4	Size	Get	UINT	アトリビュート3のバイト数

下記も参照してください。

[ネットワークのデータ交換, ページ 14](#)

[EtherNet/IPホストオブジェクト \(F8h\), ページ 175](#) (インスタンスアトリビュート#7)

インスタンス96hアトリビュート (Consuming インスタンス)

このインスタンスのインスタンス番号は、対応するアトリビュートをEtherNet/IPホストオブジェクトに実装すると変更できます。

#	名前	アクセス	タイプ	値/説明
3	Produced Data	Set	BYTEの配列	このデータはリードプロセスデータに対応します。
4	Size	Get	UINT	アトリビュート3のバイト数

下記も参照してください。

[ネットワークのデータ交換, ページ 14](#)

[EtherNet/IPホストオブジェクト \(F8h\), ページ 175](#) (インスタンスアトリビュート#8)

10.6 コネクションマネージャ (06h)

カテゴリ

拡張

オブジェクトの説明

このオブジェクトは、複数のサブネットに渡って開設されているコネクションを含む、接続及び非接続通信の為に使用されます。

サポートされているサービス

クラス : -

インスタンス :

- Get Attribute All
- Get Attribute Single
- Set Attribute Single
- Large_Forward_Open
- Forward_Open
- Forward_Close
- Unconnected Send (未接続ルーティングが有効なとき)

クラスアトリビュート

(対応クラスアトリビュートなし)

インスタンスアトリビュート

#	名前	アクセス	タイプ	値/説明
1	Open Requests	Set	UINT	受信されたForward Openサービス要求数。
2	Open Format Rejects	Set	UINT	フォーマット不正により拒否されたForward Openサービス要求数。
3	Open Resource Rejects	Set	UINT	リソース不足により拒否されたForward Openサービス要求数。
4	Open Other Rejects	Set	UINT	フォーマット不正およびリソース不足以外の原因が理由で拒否されたForward Openサービス要求数。
5	Close Requests	Set	UINT	受信されたForward Closeサービス要求数。
6	Close Format Rejects	Set	UINT	フォーマットが不正なため拒否されたForward Closeサービス要求数。
7	Close Other Rejects	Set	UINT	フォーマット不正以外の理由で拒否されたForward Openサービス要求数。
8	Connection Timeouts	Set	UINT	このコネクションマネージャによって制御されている接続中に発生した接続タイムアウトの合計数。

クラス0コネクションの詳細

概要

Class 0コネクションはセーフティ接続に対してのみサポートされます。Anybus CompactComデバイスはセーフティ接続に対して透過的なブリッジとして機能します。即ち、セーフティ接続とセーフティ I/Oデータに対するオープンとクローズ要求がセーフティモジュールへ転送されることを意味します。クラス0コネクションはUDP伝送を使用します。

サポートされているクラス0
コネクションの合計数: 2

最大入力コネクションサイズ: 241バイト
(モードバイト、現状、補完、タイムスタンプセクションを含む。)

最大出力コネクションサイズ: 239バイト
(モードバイト、現状、補完、タイムスタンプセクションを含む。)

サポートされているRPI
(Requested Packet Interval): 1...20000 ms

Class 1接続の詳細

概要

Class 1コネクションはI/Oデータの転送のために使用され、アセンブリオブジェクトにあるインスタンスに対して確立されます。各クラス1接続は二個のデータ転送を確立します。一つは消費、一つが生成です。ハートビートインスタンスは入力のみアクセスすることができるコネクションで使用されます。Class 1コネクションはUDP伝送を使用します。Null forward openに対応します。

サポートされているクラス1
コネクションの合計数: 4

最大入力コネクションサイズ: Large_Forward_Openの場合1448バイト、Forward_Openでは509バイト

最大出力コネクションサイズ: Large_Forward_Openの場合1448バイト、Forward_Openでは505バイト

サポートされているRPI
(Requested Packet Interval): 1...3200ms

T→Oコネクションタイプ: Point-to-point、Multicast、Null

O→Tコネクションタイプ: Point-to-point、Null

サポートされているトリガー
タイプ: Cyclic、CoS (Change of State)

サポートされているプライオリ
ティ: Low、High、Scheduled、Urgent

T ターゲット、この場合はモジュール

O オリジン、この場合はマスター

コネクションタイプ

- **Exclusive-Owner**コネクション

このタイプのコネクションはAnybusモジュールの出力を制御し、その他のコネクションに依存しません。

Exclusive-Ownerコネクションの最大数: 1
大数:

コネクションポイントO→T: アセンブリオブジェクト、インスタンス96h (デフォルト)

コネクションポイントT→O: アセンブリオブジェクト、インスタンス64h (デフォルト)

- **Input-Only**コネクション

このタイプの接続は出力を制御せず、データをAnybusモジュールから読み出すのに使用されます。その他の接続に依存しません。

Input-Onlyコネクションの最大数: 最大4
(Exclusive-OwnerとListen-Onlyコネクションで共有)

コネクションポイントO→T: アセンブリオブジェクト、インスタンス03h (デフォルト)

コネクションポイントT→O: アセンブリオブジェクト、インスタンス64h (デフォルト)

Exclusive-Ownerコネクションがモジュールに対して開かれてタイムアウトすると、Input-Onlyコネクションもタイムアウトします。Exclusive-Ownerコネクションが正しく閉じられても、Input-Onlyコネクションは影響を受けません。

- **拡張Input-Only**コネクション

このコネクションの機能は標準のInput-Onlyコネクションと同じです。しかし、このコネクションがタイムアウトしてもアプリケーションの状態に影響しません。

コネクションポイントO→T: アセンブリオブジェクト、インスタンス06h (デフォルト)

コネクションポイントT→O: アセンブリオブジェクト、インスタンス64h (デフォルト)

- **Listen-Only**コネクション

このタイプのコネクションが存在するために、もう一つの接続が必要です。そのコネクション (Exclusive-OwnerまたはInput-Only) が閉じられると、Listen-Onlyコネクションも閉じられます。

Listen-Onlyコネクションの最大数: 最大4
(Exclusive-OwnerとInput-Onlyコネクションで共有)

コネクションポイントO→T: アセンブリオブジェクト、インスタンス04h (デフォルト)

コネクションポイントT→O: アセンブリオブジェクト、インスタンス64h (デフォルト)

- **拡張Listen-Only**コネクション

このコネクションの機能は標準のListen-Onlyコネクションと同じです。しかし、このコネクションがタイムアウトしてもアプリケーションの状態に影響しません。

コネクションポイントO→T: アセンブリオブジェクト、インスタンス07h (デフォルト)

コネクションポイントT→O: アセンブリオブジェクト、インスタンス64h (デフォルト)

- **Redundant-Owner**コネクション

このコネクションタイプはモジュールによってサポートされていません。

Class 3 接続の詳細

概要

Class 3コネクションはメッセージルーターとの接続を確立するために使用されます。その後、接続はexplicitメッセージのために使用されます。クラス3コネクションはTCP伝送を使用します。

クラス3同時コネクション数:	6
サポートされているRPI (Requested Packet Interval):	100...10000 ms
T→Oコネクションタイプ:	Point-to-point
O→Tコネクションタイプ:	Point-to-point
サポートされているトリガータイプ:	Application
サポートされている接続サイズ:	1526バイト

10.7 パラメータオブジェクト (0Fh)

カテゴリ

拡張

オブジェクトの説明

パラメータオブジェクトはモジュールの設定データとのインターフェースになります。これは各モジュール設定パラメータ、ならびに各パラメータ (最小値と最大値、パラメータを記述する文字列を含む) の完全な記述を定義し、記述するために必要なすべての情報を提供します。RSNetworxなどの設定ツールはアプリケーション データ インスタンス (ADIs) についての情報を抽出し、ユーザーに対してはこれらの実際の名前と範囲を表します。

このプロセスには若干時間がかかるため (特にシリアルホストインターフェースを使用する場合)、EtherNet/IP ホストオブジェクトでこの機能への対応を無効にすることができます。

各パラメータは一個のインスタンスによって表されます。インスタンス番号は1から始まり、増分は1ずつ、リストにギャップはありません。CIP規格による制限があるため、このオブジェクトを通して複数の要素 (配列など) を含むADIを表すことはできません。そのような場合デフォルト値が返されます。

下記も参照してください。

- [ADIオブジェクト \(A2h\), ページ 102](#) (CIPオブジェクト)
- [EtherNet/IPホストオブジェクト \(F8h\), ページ 175](#) (ホストアプリケーションオブジェクト)

サポートされているサービス

クラス :	Get_Attribute_Single
インスタンス :	Get_Attribute_Single Set_Attribute_Single Get_Attributes_All Get_Enum_String

クラスアトリビュート

#	名前	アクセス	タイプ	値										
1	Revision	Get	UINT	0001h (オブジェクトのリビジョン)										
2	Max instance	Get	UINT	生成されるインスタンスの最大番号 = アプリケーションデータオブジェクトのクラスアトリビュート3 (『Anybus CompactCom 40 Software Design Guide』を参照)										
8	Parameter Class Descriptor	Get	WORD	デフォルト: 0000 0000 0000 1011b <table border="1"> <tr> <th>ビット:</th> <th>内容:</th> </tr> <tr> <td>0</td> <td>パラメータ インスタンスをサポート</td> </tr> <tr> <td>1</td> <td>フルアトリビュートをサポート</td> </tr> <tr> <td>2</td> <td>不揮発性ストレージのsaveコマンドを実行しなければなりません</td> </tr> <tr> <td>3</td> <td>パラメータは不揮発ストレージに保存されます</td> </tr> </table>	ビット:	内容:	0	パラメータ インスタンスをサポート	1	フルアトリビュートをサポート	2	不揮発性ストレージのsaveコマンドを実行しなければなりません	3	パラメータは不揮発ストレージに保存されます
ビット:	内容:													
0	パラメータ インスタンスをサポート													
1	フルアトリビュートをサポート													
2	不揮発性ストレージのsaveコマンドを実行しなければなりません													
3	パラメータは不揮発ストレージに保存されます													
9	Configuration Assembly Instance	Get	UINT	0000h (アプリケーションによるコンフィグレーションデータのサポートなし) 0005h (アプリケーションがコンフィグレーションデータに対応する場合。但し、コンフィグレーションインスタンス番号がEtherNet/IP ホストオブジェクトのアトリビュート 15を使用して変更されている場合を除く。)										

インスタンスアトリビュート

#	名前	アクセス	タイプ	値/説明
1	Parameter Value	Get/Set	アトリビュート4、5、6で指定される。	パラメータの実際値 このアトリビュートは、アトリビュート#4のビット4がTrueの場合は読み取り専用
2	Link Path Size	Get	USINT	0007h (バイト単位のリンクパスサイズ)
3	Link Path	Get	Packed EPATH	20 A2 25 nn nn 30 05h (このパラメータの値が取得されるオブジェクトへのパス。この場合はADIオブジェクト)
4	Descriptor	Get	WORD	ビット: 内容:

#	名前	アクセス	タイプ	値/説明
				0 設定可能なバスをサポート(N/A) 1 Enum文字列をサポート 2 スケーリングをサポート(N/A) 3 スケーリングリンクをサポート(N/A) 4 読取り専用パラメータ 5 モニターパラメータ (N/A) 6 拡張精度スケーリングをサポート(N/A)
5	データ型	Get	USINT	データ型を表すコード
6	Data Size	Get	USINT	パラメータ値のバイト数
7	Parameter Name String	Get	SHORT_STRING	パラメータ名、16文字で切り捨て
8	Units String	Get	SHORT_STRING	「」 (デフォルト文字列)
9	Help String	Get	SHORT_STRING	
10	Minimum Value	Get	(データ型)	パラメータの最小値 データ型はアトリビュート5で定義されます。
11	Maximum Value	Get	(データ型)	パラメータの最大値 データ型はアトリビュート5で定義されます。
12	デフォルト値	Get	(データ型)	パラメータのデフォルト値 データ型はアトリビュート5で定義されます。
13	Scaling Multiplier	Get	UINT	0001h
14	Scaling Divisor	Get	UINT	
15	Scaling Base	Get	UINT	
16	Scaling Offset	Get	INT	
17	Multiplier Link	Get	UINT	
18	Divisor Link	Get	UINT	0000h
19	Base Link	Get	UINT	
20	Offset Link	Get	UINT	
21	Decimal Precision	Get	USINT	
				00h

デフォルト値

#	名前	値	コメント
1	Parameter Value	0	-
2	Link Path Size	0	バイト単位のリンクパスサイズ
3	Link Path	-	NULL/パス
4	Descriptor	0010h	読取り専用パラメータ
5	Data type	C6h	USINT
6	Data size	1	-
7	Parameter Name String	「予約」	-
8	Units String	「」	-
9	Help String	「」	-
10	Minimum value	N/A	0
11	Maximum value	N/A	0
12	Default value	N/A	0
13	Scaling Multiplier	N/A	1
14	Scaling Divisor	N/A	1
15	Scaling Base	N/A	1
16	Scaling Offset	N/A	0
17	Multiplier Link	N/A	0
18	Divisor Link	N/A	0
19	Base Link	N/A	0
20	Offset Link	N/A	0
21	Decimal Precision	N/A	0

10.8 Time Syncオブジェクト (43h)

カテゴリ

拡張

オブジェクトの説明

Time SyncオブジェクトはIEEE 1588 (IEC 61588) 規格のPrecision Time Protocol (PTP) にCIPインターフェースを提供します。このプロトコルは、分散型デバイスのネットワーク全体に、クロック同期のための標準的なメカニズムを提供します。CIP Syncはデフォルトで無効になっていますが、EtherNet/IPホストオブジェクト (F8h) でアトリビュート#32を設定すると有効になります。

下記も参照してください。

- [QoSオブジェクト \(48h\), ページ 97](#) (CIPオブジェクト)
- [EtherNet/IPホストオブジェクト \(F8h\), ページ 175](#) (ホストアプリケーションオブジェクト)

サポートされているサービス

クラス : Get_Attribute_Single

インスタンス : Get_Attribute_Single
Set_Attribute_Single
Get_Attributes_List
Set_Attributes_List

クラスアトリビュート

#	名前	アクセス	タイプ	値
1	Revision	Get	UINT	0004h (オブジェクトリビジョン)

インスタンスアトリビュート

#	名前	アクセス	タイプ	値	説明
1	PTPEnable	Get/Set	BOOL	-	0 = 無効 1 = 有効 (デフォルト)
2	IsSynchronized	Get	BOOL	-	0 = 非同期 1 = 同期
3	SystemTimeMicroseconds	Get	ULINT	-	system_timeの現在値 (μs)
4	SystemTimeNanoseconds	Get	ULINT	-	ナノ秒 (ns) で示されるsystem_timeの現在値
5	OffsetFromMaster	Get	LINT	-	ローカルクロックとマスタークロック間のオフセット
6	MaxOffsetFromMaster	Get/Set	ULINT	-	ローカルクロックとマスタークロック間の最大オフセット。モジュールによる監視クリアするには0を設定
7	MeanPathDelayToMaster	Get	LINT	-	マスターに対する平均パス遅延

#	名前	アクセス	タイプ	値	説明
8	GrandMasterClockInfo	Get	構造体:		グランドマスタークロック情報
	ClockIdentity		USINT[8]	-	クロックID
	ClockClass		UINT	-	クロッククラス
	TimeAccuracy		UINT	-	時間精度
	OffsetScaledLogVariance		UINT	-	オフセットスケールリング付きの対数分散
	CurrentUtcOffset		UINT	-	現在のUTCオフセット
	TimePropertyFlags		WORD	-	時間プロパティのフラグ
	TimeSource		UINT	-	タイムソース
	Priority1		UINT	-	Priority1
	Priority2		UINT	-	Priority2
9	ParentClockInfo	Get	構造体:		親クロック情報
	ClockIdentity		USINT[8]	-	クロックID
	PortNumber		UINT	-	PTPポート番号
	ObservedOffsetScaledLogVariance		UINT	-	観察されたオフセットスケールリング付きの対数分散
	ObservedPhaseChangeRate		UDINT	-	観察された位相変化率
10	LocalClockInfo	Get	構造体:		ローカルクロック情報
	ClockIdentity		USINT[8]	-	クロックID
	ClockClass		UINT	-	クロッククラス
	TimeAccuracy		UINT	-	時間精度
	OffsetScaledLogVariance		UINT	-	オフセットスケールリング付きの対数分散
	CurrentUtcOffset		UINT	-	現在のUTCオフセット
	TimePropertyFlagsNone		WORD	-	時間プロパティのフラグ
	TimeSource		UINT	-	タイムソース
11	NumberOfPorts	Get	UINT	1	デバイスは、通常のクロックとエンドツーエンドのトランスペアレントクロックを持つ、ハイブリッドクロックを備えています。
12	PortStateInfo	Get	構造体:		PTPポートの状態に関する情報
	NumberOfPorts		UINT	1	PTPポートの数。 アトリビュート#11の説明を参照のこと
	PortNumber		UINT	-	PTPポート番号
	PortState		UINT	-	PTPポートの状態
13	PortEnableCfg	Get/Set	構造体:		PTPポート有効化設定
	NumberOfPorts		UINT	1	PTPポートの数。 アトリビュート#11の説明を参照のこと
	PortNumber		UINT	-	PTPポート番号
	PortEnable		UINT	-	0 = 無効 1 = 有効
14	PortLogAnnounceIntervalCfg	Get/Set	構造体:		PTPポートのログアナウンス間隔設定
	NumberOfPorts		UINT	1	PTPポートの数。 アトリビュート#11の説明を参照のこと
	PortNumber		UINT	-	PTPポート番号
	PortLogAnnounceInterval		UINT	-	PTPポートのログアナウンス間隔

#	名前	アクセス	タイプ	値	説明
15	PortLogSyncIntervalCfg	Get/Set	構造体:		PTPポートのログ同期間隔設定
	NumberOfPorts		UINT	-	PTPポートの数。 アトリビュート#11の説明を参照のこと
			構造体の配列:		
	PortNumber		UINT	-	PTPポート番号
	PortLogSyncInterval		INT	-	PTPポートのログ同期間隔
18	DomainNumber	Get/Set	USINT	-	ドメイン番号
19	ClockType	Get	WORD	-	クロックタイプ
20	ManufactureIdentity	Get	USINT[4]	-	最初の3オクテットがMACアドレスの最初の3オクテットを保持します。4つ目のオクテットは予約されています (0に設定)。 デフォルト: [0x00, 0x30, 0x11, 0x00]
21	ProductDescription	Get	構造体:		ベンダー名、製品名、シリアル番号 (16進数表記)が含まれます。記述の最大長は64バイトです。シリアル番号とセミコロンで10バイトを占めます。ベンダー名と製品名を合わせた長さが54バイトを超える場合は、最大長64バイトを超えないようにベンダー名が切り捨てられます。
	Size		UDINT	10 ~ 64	
	説明		USINTの配列	「<vendor name>;<product name>;<serial number>」	
22	RevisionData	Get	構造体:		リビジョンデータ
	Size		UDINT	-	リビジョンデータのサイズ
	Revision		USINTの配列	<HDL clock version>;<clock fw component version>;<clock sw component version>	Revision
23	UserDescription	Get/Set	構造体:		フォーマット: <device name>;<location>
	Size		UDINT	1	
	説明		USINTの配列	「」	
24	PortProfileIdentityInfo	Get	構造体:		PTPポートのプロファイルID情報
	NumberOfPorts		UDINT	1	PTPポートの数 アトリビュート#11の説明を参照のこと
			構造体の配列:		
	PortNumber		UINT	-	PTPポート番号
	PortProfileIdentity		USINT[8]	-	PTPポートのプロファイルID
25	PortPhysicalAddressInfo	Get	構造体:		PTPポートの物理アドレス情報
	NumberOfPorts		UINT	1	PTPポートの数 アトリビュート#11の説明を参照のこと
			構造体の配列:		
	PortNumber		UINT	-	PTPポート番号
	PhysicalProtocol		USINT[16]	-	物理プロトコル
	SizeOfAddress		UINT	-	アドレスのサイズ
	PortPhysicalAddress		USINT[16]	-	PTPポートの物理アドレス

#	名前	アクセス	タイプ	値	説明	
26	PortProtocolAddressInfo	Get	構造体:		PTPポートのプロトコルアドレス情報	
	NumberOfPorts		UINT	1	PTPポートの数 アトリビュート#11の説明を参照のこと	
			構造体の配列:	-		
	PortNumber		UINT	-	PTPポート番号	
	NetworkProtocol		UINT	-	ネットワークプロトコル	
	SizeOfAddress		UINT	-	アドレスのサイズ	
	PortProtocolAddress		USINT[16]	-	PPTポートのプロトコルアドレス	
27	StepsRemoved	Get	UINT	-	削除されたステップ	
28	SystemTimeAndOffset	Get	構造体:		システムの時間およびオフセット	
	SystemTime		ULINT	-	システム時間	
	SystemOffset		ULINT	-	システムオフセット	
29	Associated Interface Objects	Get	構造体:		このアトリビュートは、CIPポートオブジェクト (F4h) がホストアプリケーションによって有効にされている場合にのみ実装され、デバイスの EtherNet/IPポートを表すCIPポートインスタンス#1をPTPポートに関連付けます。	
	Number of Ports		UINT	1		
			構造体の配列:			
	Port Number		UINT	1		
	Associated Object Path Size		USINT	2		
	Associated Object		Padded EPATH	[0x20、0xF4、0x24、0x01]		

10.9 DLRオブジェクト (47h)

カテゴリ

拡張

オブジェクトの説明

デバイスレベルリング (DLR)オブジェクトはDLR プロトコル用のステータス情報インターフェースになります。このプロトコルはEthernet リングトポロジの使用を可能にし、DLRオブジェクトはプロトコルとのCIP アプリケーションレベルでのインターフェースを提供します。

このオブジェクトは、DLRがEtherNet/IPホストオブジェクトで無効にされているときは利用できません ([Ethernet ホストオブジェクト \(F9h\)](#), ページ 184を参照)。

サポートされているサービス

クラス : Get_Attribute_Single
 Get_Attributes_All
インスタンス : Get_Attribute_Single

クラスアトリビュート

#	名前	アクセス	タイプ	値
1	Revision	Get	UINT	0003h (オブジェクトリビジョン)

インスタンスアトリビュート

アトリビュート#1~4と#6~7はEtherNet/IPホストオブジェクトを実装することによってカスタム化されます ([EtherNet/IPホストオブジェクト \(F8h\)](#), ページ 175を参照)。

#	名前	アクセス	タイプ	値/説明
1	Network Topology	Get	USINT	ビット: 内容: 0 「リニア」 1 「リング」
2	Network Status	Get	USINT	ビット: 内容: 0 「ノーマル」 (N/A) 1 「リング障害」 2 「予期されないループ検出」 3 「ネットワークの一部障害」 4 「障害/復旧の高速周期」
10	Active Supervisor Address	Get	構造体: UDINT 以下の配列: 6 USINTs	このアトリビュートは、アクティブなリングスーパーバイザのIPアドレス (IPv4) および/またはイーサネットMacアドレスを保持します。
12	Capability Flags	Get	DWORD	82h (ビーコンベースリングノード、Flush_Tableフレーム対応)

10.10 QoS オブジェクト (48h)

カテゴリ

拡張

オブジェクトの説明

サービス品質(QoS)は異なる相対プライオリティやその他のデリバリ特性があるトラフィックの流れを取り扱うように機能するメカニズムに適用される一般用語です。標準的なQoSメカニズムにはTCP/IP プロトコルスイート上でのIEEE 802.1D/Q (Ethernetフレームプライオリティ)と分化サービス(DiffServ)が含まれます。

QoSオブジェクトはEtherNet/IP デバイスにおける一部のQoS関連動作を構成するための手段となります。

このオブジェクトはノンゼロDiffServコードポイント (DSCP)を持つEtherNet/IPメッセージを送ることや802.1Qタグ化フレームでEtherNet/IPメッセージを送ることに対応するデバイスのために必要とされます。

サポートされているサービス

クラス : Get_Attribute_Single
 インスタンス : Get_Attribute_Single
 Set_Attribute_Single

クラスアトリビュート

#	名前	アクセス	タイプ	値
1	Revision	Get	UINT	0001h (オブジェクトのレビジョン)

インスタンスアトリビュート

アトリビュート#1~4と#6~7はEtherNet/IPホストオブジェクトを実装することによってカスタム化されます ([EtherNet/IPホストオブジェクト \(F8h\)](#), ページ 175を参照)。

#	名前	アクセス	タイプ	値/説明
1	802.1Q Tag Enable	Set	USINT	802.1Qフレーム送信の有効化・無効化。 値: 内容: 0 無効 (デフォルト) 1 有効
2	DSCP PTP Event	Set	USINT	PTP (IEEE 1588) イベントメッセージのDSCP値。 デフォルト値 = 59
3	DSCP PTP General	Set	USINT	PTP (IEEE 1588) 一般メッセージのDSCP値。 デフォルト値 = 47
4	DSCP Urgent	Set	USINT	プライオリティがUrgentのCIP伝送クラス1メッセージ デフォルト: 55
5	DSCP Scheduled	Set	USINT	プライオリティがScheduledのCIP伝送クラス1メッセージ デフォルト: 47
6	DSCP High	Set	USINT	プライオリティがHighのCIP伝送クラス1メッセージ デフォルト: 43
7	DSCP Low	Set	USINT	プライオリティがLowのCIP伝送クラス1メッセージ デフォルト: 31
8	DSCP Explicit	Set	USINT	CIP UCMMとCIPクラス3 デフォルト: 27

10.11 Base Energy オブジェクト (4Eh)

カテゴリ

拡張

オブジェクトの説明

ベースエネルギーオブジェクトは、CIPエネルギー実装の「エネルギースーパーバイザ」として機能します。エネルギー値のタイムベースやエネルギーモードサービスを提供し、産業施設の様々なレベルまでエネルギー値を集約するための集約サービスを提供することができます。これはさらにエネルギー計測の結果を表すための標準フォーマットも提供します。このオブジェクトはエネルギーの種類から独立的であり、エネルギー種別のデータと機能がエネルギーシステムに標準的方法で統合されるようにします。Anybus CompactCom 40 EtherNet/IPはベースエネルギーオブジェクトの1つのインスタンスをサポートします。例えば、電力モニターは別個の電力メーターのメーターリングパルス出力の推移を計数することができます。ベースエネルギーオブジェクトインスタンスによって表されるこのような推移の計数は、別個のメーターによって測定された電力消費量を反映します。

ベースエネルギーオブジェクトインスタンスはスタンドアロンインスタンスとして、または電力ないし電力以外のエネルギーオブジェクトインスタンスと並列的に存在できます(これらのオブジェクトはAnybus CompactCom 40 EtherNet/IPには実装されていません)。これらのオブジェクトのインスタンスがデバイスに実装されている場合、このインスタンスがデバイスのベースエネルギーオブジェクトインスタンスと関連付けられていなければなりません。

このオブジェクトがネットワークにアクセスできるためには、エネルギーレポートオブジェクト(E7h)がホストアプリケーションに実装されていなければなりません ([Energy Reporting オブジェクト \(E7h\)](#), ページ 162を参照)。

サポートされているサービス

クラス : Get_Attribute_Single

インスタンス : Get_Attribute_Single

クラスアトリビュート

#	名前	アクセス	タイプ	値
1	Revision	Get	UINT	0002h (オブジェクトリビジョン)

インスタンスアトリビュート

アトリビュート#1~4と#6~7はEtherNet/IPホストオブジェクトを実装することによってカスタム化されます ([EtherNet/IPホストオブジェクト \(F8h\)](#), ページ 175を参照)。

#	名前	アクセス	タイプ	値/説明
1	Energy/Resource Type	Get	UINT	このインスタンスによって管理されるエネルギー種別 常に0 (ジェネリック)
2	Base Energy Object Capabilities	Get	UINT	常に0 (エネルギー測定あり)
3	Energy Accuracy	Get	UINT	0.01% (デフォルト) またはアトリビュート#4で指定されているその他の単位における0.01相当分のいずれかの単位で、電力とエネルギー計測結果の正確さを指定します。0の場合、不明。
4	Energy Accuracy Basis	Get	UINT	常に0 (読取り値のパーセント)
7	Consumed Energy Odometer	Get	ODOMETER	消費されたエネルギーの値。
8	Generated Energy Odometer	Get	ODOMETER	生成されたエネルギーの値。
12	Energy Type Specific Object Path	Get	構造体: UINT (パスサイズ) Padded EPATH (パス)	NULLパス

- インスタンスが消費エネルギーか生産エネルギーをレポートするかに応じて、アトリビュート #7またはアトリビュート #8は必須です。
- 構造体データ型ODOMETERで莫大な値を表すことが可能になります。詳細についてはCIP仕様Volume 1 (CIP Common)を参照してください。

10.12 Power Management オブジェクト (53h)

カテゴリ

拡張

オブジェクトの説明

Power Management オブジェクトは一時停止またはスリープ状態に出入りするデバイスの制御をサポートする為に、標準化アトリビュートとサービスを提供します。このオブジェクトがネットワークにアクセスできるためには、Energy Control オブジェクト (F0h)が実装されなければなりません。

以下も参照 ..

- Energy Control オブジェクト (F0h) (『Anybus CompactCom 40 Software Design Guide』)

サポートされているサービス

クラス : Get_Attribute_Single
インスタンス : Get_Attribute_Single
Power_Management
Set_Pass_Code
Clear_Pass_Code

クラスアトリビュート

#	名前	アクセス	タイプ	値
1	Revision	Get	UINT	0002h (オブジェクトリビジョン)

インスタンスアトリビュート

#	名前	アクセス	タイプ	値/説明
1	Power Management Command	Get	DWORD	最近の電力管理要求から成る各種ビットフィールド。
2	Power Management Status	Get	DWORD	電力管理状態の情報を提供する各種ビットフィールド。
3	Client Path	Get	構造体:	このインスタンス (サーバ) から現在のオーナー (クライアント) へのEPATHを指定します。
			UINT (パスサイズ)	パスサイズ (ワード)
			Padded EPATH (ノバス)	
4	Number of Power Management Modes	Get	UINT	アトリビュート5にある電力管理モード配列エントリ数。
5	Power Management Nodes	Get	以下の配列:	低電力モードの配列
			構造体:	モード (モード構造体の配列)
			USINT	Minimum Pause Units (最小一時停止時間の単位を指定)
			UINT	最小一時停止時間
			USINT	Resume Units (再開時間の単位を指定)
			UINT	Resume Time (一時停止状態とされたときからオーナー状態までの推移に必要な所要時間)
			REAL	Power Level (このモードにおけるkW単位の電力)
BOOL	Availability (現在のデバイス状態を前提にしてこのモードに入れるかどうかを指定)			
6	Sleeping State Support	Get	BOOL	0 (スリープ状態のサポートなし)

10.13 ADIオブジェクト (A2h)

カテゴリ

拡張

オブジェクトの説明

このオブジェクトはアプリケーション データオブジェクトのインスタンスをEtherNet/IPにマッピングします。このオブジェクトに対するすべての要求はホストアプリケーションにあるアプリケーション データオブジェクトに対する明示的なオブジェクト要求に変換されます。次に応答がCIPフォーマットに戻し変換され、オリジネータへ送られます。

ADIオブジェクトはEtherNet/IP ホストオブジェクト (F8h)のアトリビュート 30を使って無効にできます。このアトリビュートでADIオブジェクト番号の変更もできます。

以下も参照 ..

- アプリケーション データオブジェクト (『Anybus CompactCom 40 Software Design Guide』 参照)
- [パラメータオブジェクト \(0Fh\), ページ 89](#) (CIPオブジェクト)
- [EtherNet/IPホストオブジェクト \(F8h\), ページ 175](#)

サポートされているサービス

クラス : Get_Attribute_Single

インスタンス : Get_Attribute_Single

Set_Attribute_Single

クラスアトリビュート

#	名前	アクセス	タイプ	値
1	Revision	Get	UINT	0002h (オブジェクトリビジョン)
2	Max Instance	Get	UINT	アプリケーションデータオブジェクトのアトリビュート#4と同義
3	Number of instances	Get	UINT	アプリケーションデータオブジェクトのアトリビュート#3と同義

アプリケーションデータオブジェクトに関する情報は『Anybus CompactCom 40 Software Design Guide』を参照してください。

インスタンスアトリビュート

各インスタンスはアプリケーションデータオブジェクト内のインスタンスに対応します (詳細は、全般的な『Anybus CompactCom 40 Software Design Guide』を参照してください)。

#	名前	アクセス	タイプ	値/説明
1	Name	Get	SHORT_STRING	パラメータ名 (長さを含む)
2	ABCC Data type	Get	USINTの配列	インスタンス値のデータ型
3	No. of Elements	Get	USINT	指定したデータ型の要素数
4	Descriptor	Get	USINTの配列	このインスタンスへのアクセス権を表すビットフィールド ビット: 意味: 0 1 = Getアクセス 1 1 = Setアクセス 2 (予約、0に設定) 3 1 = ライトプロセスデータのマッピングが可能 4 1 = リードプロセスデータのマッピングが可能 5 1 = NVSパラメータ 6 1 = データ通知が有効
5	値	Get/Set	アトリビュート#2、 #3、#9によって決定	インスタンス値
6	Max Value	Get		最大許容パラメータ値。
7	Min Value	Get		最小許容パラメータ値。
8	デフォルト値	Get		デフォルトパラメータ値。
9	Number of subelements	Get	UINTの配列	ADIのサブエレメント数。アプリケーションに未実装の場合はデフォルト値は1です。 配列のサイズはアトリビュート#3に依存します。

アトリビュート#5～8はモジュールによってCIP規格と双方向に変換されます。

10.14 ポートオブジェクト(F4h)

カテゴリ

拡張

オブジェクトの説明

ポートオブジェクトはデバイスに存在するCIPポートを記述します。各ルーティング可能なCIPポートは個別のインスタンスに記述されます。ルーティング不可のポートを記述することもできます。このオブジェクトをサポートするために単一CIPポートを持つデバイスは不要です。

オブジェクトはEtherNet/IPホストオブジェクトで有効にされている場合しか存在しません(インスタンスアトリビュート#17)。

以下も参照 ..

- [EtherNet/IPホストオブジェクト \(F8h\), ページ 175](#) (Anybusモジュールオブジェクト)
- [CIPポートコンフィグレーションオブジェクト \(0Dh\), ページ 152](#) (ホストアプリケーションオブジェクト)

サポートされているサービス

クラス :	Get_Attributes_All Get_Attribute_Single
インスタンス :	Get_Attributes_All Get_Attribute_Single

クラスアトリビュート

#	名前	アクセス	タイプ	値
1	Revision	Get	UINT	0002h (オブジェクトリビジョン)
2	Max Instance	Get	UINT	最大インスタンス番号
3	Number of Instances	Get	UINT	現在作成されているポート数。
8	Entry Port	Get	UINT	この要求がデバイスに入るとき通過したポートを記述するポートオブジェクトのインスタンスを返します。
9	Port Instance Info	Get	以下の配列:	各インスタンスのインスタンスアトリビュート1と2を格納する構造体の配列。配列には最大インスタンス数までのインスタンス番号によるインデックスが付きます。インデックス1 (オフセット0) と任意の非インスタンス化インスタンスにおける値はゼロになります。
			構造体: UINT (タイプ) UINT (番号)	ポート種別を列挙 (インスタンスアトリビュート#1を参照) このポートに関連付けられているCIPポート番号 (インスタンスアトリビュート#2を参照)

インスタンスアトリビュート (インスタンス#1)

#	名前	アクセス	タイプ	値/説明
1	Port Type	Get	UINT	0h (デフォルト) 4h (アプリケーションがポートを登録した場合)
2	Port Number	Get	UINT	2h
3	Link Object	Get	構造体: UINT Padded EPATH	- 2h (パス長) 20 F5 24 01h (リンクパス)
4	Port Name	Get	SHORT_STRING	「EtherNet/IP」
5	Port Type Name	Get	SHORT_STRING	「」
6	Port Description	Get	SHORT_STRING	「」
7	Node Address	Get	Padded EPATH	-
10	Port Routing Capabilities	Get	UDINT	1h (未接続着信メッセージのルーティングのサポートあり)

下記も参照してください。

[CIPポートコンフィグレーションオブジェクト \(0Dh\), ページ 152](#)

インスタンスアトリビュート (インスタンス#2 ~ #8)

#	名前	アクセス	タイプ	値/説明
1	Port Type	Get	UINT	ポートタイプを列挙します
2	Port Number	Get	UINT	このポートと関連付けられたCIPポート番号
3	Link Object	Get	構造体: UINT Padded EPATH	- Path length (16ビット語長) このポートに対するオブジェクトを識別する論理パスセグメント。パスは1つの論理クラスセグメントと1つの論理インスタンスセグメントから構成されなければなりません。最大サイズは12バイトです。
4	Port Name	Get	SHORT_STRING	「ポートA」などのポート名。最大64文字。
5	Port Type Name	Get	SHORT_STRING	「」
6	Port Description	Get	SHORT_STRING	「」
7	Node Address	Get	Padded EPATH	ポートにおけるこのデバイスのノード番号。このデータ型での範囲はポートセグメントまでに限定されます。
8	Port Node Range	Get	構造体: UINT (最小) UINT (最大)	- ポートでの最小ノード番号 ポートでの最大ノード番号
10	Port Routing Capabilities	Get	UDINT	1h (未接続着信メッセージのルーティングのサポートあり)

下記も参照してください。

[CIPポートコンフィグレーションオブジェクト \(0Dh\), ページ 152](#)、「インスタンスアトリビュート」

10.15 TCP/IPインターフェースオブジェクト (F5h)

カテゴリ

拡張

オブジェクトの説明

このオブジェクトはモジュールのTCP/IPネットワークインターフェースを構成するためのメカニズムを提供します。これはTCP/IP関連の全設定をTCP/IP対応の各通信インターフェースに対して1つのインスタンスにグループ化します。

以下も参照してください。

- [通信設定, ページ 12](#)
- [ネットワークコンフィグレーションオブジェクト \(04h\), ページ 118](#) (Anybusモジュールオブジェクト)

サポートされているサービス

クラス : Get_Attribute_All
 Get_Attribute_Single

インスタンス : Get_Attribute_All
 Get_Attribute_Single
 Set_Attribute_Single

クラスアトリビュート

#	名前	アクセス	タイプ	値
1	Revision	Get	UINT	0004h (オブジェクトリビジョン)
2	Max instance	Get	UINT	1 (最大インスタンス番号)
3	Number of instances	Get	UINT	1 (インスタンス数)
6	Maximum ID Number Class Attributes	Get	UINT	7 (最後に実装されたクラスアトリビュートのアトリビュート番号)
7	Maximum ID Number Instance Attributes	Get	UINT	13 (最後に実装されたインスタンスアトリビュートのアトリビュート番号)

インスタンスアトリビュート

#	名前	アクセス	タイプ	値	コメント
1	状態	Get	DWORD	-	<p>ビット: 意味: (予約、0に設定)</p> <p>0-3 1hに設定されている場合、アトリビュート #5はDHCPからまたは不揮発ストレージからの有効なコンフィグレーションを格納しています。2hに設定されている場合、アトリビュート#5はハードウェア設定からの有効なコンフィグレーションを格納しています。残りの値は今後の使用のための予備です。</p> <p>4 1に設定されている場合、マルチキャストは保留中になります。</p> <p>5 1に設定されている場合、インターフェース設定は保留中になります。新たなコンフィグレーションは次のリセット時に読み込まれます。</p> <p>6 AcdStatus.アドレスの競合が検出されると1に設定されません。アドレス競合検出はアトリビュート #10で有効/無効にします。</p> <p>7 AcdFault</p> <p>8-31 (予約、0に設定)</p>
2	Configuration Capability	Get	DWORD	-	<p>ビット: 意味:</p> <p>0-1: 常に0。詳細についてはCIP仕様を参照してください。</p> <p>2: 1に設定されている場合、デバイスはDHCPクライアントとして振る舞うことができます。アトリビュート#24 (Enable DHCP Client) がEthernet ホストオブジェクト (F9h)、ページ 184で無効になっている場合、ビットは0に設定されます。</p> <p>3: 常に0。詳細についてはCIP仕様を参照してください。</p> <p>4: 'Configuration Settable' ビットは161ページの'EtherNet/IP ホストオブジェクト (F8h)'にあるインスタンスアトリビュート #9の値を反映します。</p> <p>5: このビットが1に設定されている場合、モジュールはハードウェア設定可能です。このビットは、セットアップの間にNetwork Configuration Object (04h)のいずれかのアドレスアトリビュートが設定された場合、もしくは、Application Object (FFh)のアトリビュート #6 (Hardware configurable address)がセットされている場合に、セットされます。</p> <p>6: 常に0。詳細についてはCIP仕様書をご参照ください。</p> <p>7: 1に設定されている場合、デバイスはアドレス競合を検出します。アドレス競合検出がEthernetトホストオブジェクトで無効になっている場合、ビットは0に設定されます (184ページを参照)。</p> <p>8 - 31: (予約、0に設定)</p>
3	Configuration Control	Get/Set	DWORD	-	<p>値: 意味</p> <p>0: 不揮発性メモリからの設定</p> <p>2: DHCPからの設定</p>
4	Physical Link Object	Get	構造体: UINT (パスサイズ) Padded EPATH	0002h 20 F6 24 03h	- イーサネットリンクオブジェクト、インスタンス#3へのパス
5	Interface Configuration	Get/Set	構造体: UDINT (IP) UDINT (Mask) UDINT (GW) UDINT (DNS1) UDINT (DNS2) STRING (ドメイン)		IPアドレス サブネットマスク デフォルトゲートウェイ プライマリDNS セカンダリDNS デフォルトドメイン
6	Host Name	Get/Set	STRING	-	Anybusモジュールのホスト名
8	TTL Value	Get/Set	USINT	1	EtherNet/IPマルチキャストパケット用TTL値
9	Mcast Config	Set	構造体:		IPマルチキャスト設定。
	Alloc Control		USINT	0	<p>値: 意味:</p> <p>0: マルチキャストアドレスの生成にデフォルトの割当アルゴリズムを使用</p>

#	名前	アクセス	タイプ	値	コメント
	(予約)		USINT	0	「Num Mcast」フィールドおよび「Mcast Start Addr」フィールドの値に従ってマルチキャストアドレスを割り当てます。
	Num Mcast		UINT	-1	ゼロに設定。変更しないでください。
	Mcast Start Addr		UDINT	-	EtherNet/IPに割り当てるためのマルチキャストアドレス
10	SelectAcd	Set	Bool	1	値: 意味: 0: ACDを無効化する 1: ACDを有効化 (デフォルト)。 ACD (アドレス競合検出)が有効な場合、アトリビュート #1のビット6はACD競合が検出されたときにセットされます。ネットワークステータスLEDは競合の検出も表します (前面図, ページ 192を参照)。
11	LastConflictDetected	Set	構造体:		最後に検出された競合に関連するACD診断パラメータ。
	AcdActivity		USINT	-	最後に競合が検出されたときのACDの動作状態。
	RemoteMAC		ARRAY of 6 USINT	-	競合が検出されたARPPDUからのリモートノードのMACアドレス。
	ArpPdu		ARRAY of 28 USINT	-	競合が検出された元のARP PDUのコピー。
12	EIP QuickConnect	Set	Bool	0	値: 意味: 0: EIP QuickConnectを無効化 (デフォルト) 1: EIP QuickConnectを有効化 EIP QuickConnectが有効になっていると、QuickConnect機能はEtherNet/IPのターゲットデバイスを高速電源投入し、EtherNet/IPネットワークに接続させます。
13	Encapsulation inactivity timeout	Set	UINT	0 - 3600	TCP接続が閉じられる前の無動作秒数。 0: 無効

- ネットワークからのネットワーク設定 (アトリビュート#3、#5) のコンフィギュレーションを可能にするサポートは、アトリビュート#9をEtherNet/IPホストオブジェクトに実装することによって無効にできます ([EtherNet/IPホストオブジェクト \(F8h\)](#), ページ 175を参照)。
- アトリビュート #10と#11は、Ethernet ホストオブジェクト (F9h)のアトリビュート #11を使用してACDが無効にされた場合には利用できなくなります。
- アトリビュート#12:
 - モジュールがEIP QuickConnect 機能を使用するように構成されている場合、EDSファイルを変更しなければなりません。EDSファイルが変更されたら、モジュールのアイデンティティも変更し、モジュールの認証が必要になります。
 - このアトリビュートは、EtherNet/IPホストオブジェクトにアトリビュート#26が実装されている場合に存在しません ([EtherNet/IPホストオブジェクト \(F8h\)](#), ページ 175を参照)。

10.16 Ethernet リンクオブジェクト (F6h)

カテゴリ

拡張

オブジェクトの説明

このオブジェクトは、IEEE 802.3 通信インターフェースのリンク固有カウンターとステータス情報を保持します。モジュールにおいては各通信インターフェースに対して1つのインスタンスのみサポートされます。内部でアクセス可能なインターフェースのインスタンスにも対応します。

以下も参照してください。

- [通信設定, ページ 12](#)
- [ネットワークコンフィグレーションオブジェクト \(04h\), ページ 118](#) (Anybusモジュールオブジェクト)

サポートされているサービス

```

クラス :          Get_Attributes_All
                Get_Attribute_Single

インスタンス :    Get_Attributes_All
                Get_Attribute_Single
                Set_Attribute_Single
                Get_And_Clear
  
```

クラスアトリビュート

デフォルトでは、3つのインスタンス (ポート1、ポート2、内部ポート) が実装されており、すなわちこれは2つのポートが有効であることを意味します。

Ethernetホストオブジェクト (F9h) のPort 2 Stateアトリビュートでポート2が無効になっている場合、1つのインスタンス (ポート1) しか実装できません。

#	名前	アクセス	タイプ	値
1	Revision	Get	UINT	0004h (オブジェクトリビジョン)
2	Max Instance	Get	UINT	1または3 (最大インスタンス番号)
3	Number of Instances	Get	UINT	1または3 (インスタンス数)
6	Maximum ID Number Class Attributes	Get	UINT	7 (最後に実装されたクラスアトリビュートのアトリビュート番号。)
7	Maximum ID Number Instance Attributes	Get	UINT	11 (最後に実装されたインスタンスアトリビュートのアトリビュート番号。)

インスタンスアトリビュート

#	名前	アクセス	タイプ	値	コメント
1	Interface Speed	Get	UDINT	10または100	実際のイーサネットインターフェースの速度。
2	Interface Flags	Get	DWORD	-	下表「Interface Flags」を参照してください。
3	Physical Address	Get	6 USINTの配列	(MAC ID)	物理ネットワークアドレス、すなわち割当MACアドレス。
4	Interface Counters	Get	構造体:		
	In Octets		UDINT	N/A	インターフェースで受信されたオクテット数
	In Ucast Packets		UDINT	N/A	インターフェースで受信されたユニキャストパケット数
	In NUcast Packets		UDINT	N/A	インターフェースで受信された非ユニキャストパケット数
	In Discards		UDINT	N/A	プロトコルが不明なインバウンドパケット数
	In Errors		UDINT	N/A	エラーがあるインバウンドパケット数 (In discardsは含まれません)
	In Unknown Protos		UDINT	N/A	プロトコルが不明なインバウンドパケット数
	Out Octets		UDINT	N/A	インターフェースから送られたオクテット数
	Out Ucast Packets		UDINT	N/A	インターフェースから送られたユニキャストパケット数
	Out NUcast Packets		UDINT	N/A	インターフェースから送られた非ユニキャストパケット数
	Out Discards		UDINT	N/A	プロトコルが不明なアウトバウンドパケット数
	Out Errors		UDINT	N/A	エラーがあるアウトバウンドパケット数 (Out discardsは含まれません)
	5		Media Counters	Get	構造体:
Alignment Errors		UDINT	N/A		長さが8整数以外の受信フレーム数
FCS Errors		UDINT	N/A		FCSチェックにエラーした受信フレーム数
Single Collisions		UDINT	N/A		一回だけコリジョンが発生して転送に成功したフレーム数
Multiple Collisions		UDINT	N/A		二回以上コリジョンが発生して転送に成功したフレーム数
SQE Test Errors		UDINT	0		SQEテストエラーメッセージが生成された回数 (現在のPHYインターフェースがカウンターに提供されない状態)
Deferred Transmissions		UDINT	N/A		一回目の伝送試行が媒体ビジーにより遅延しているフレーム数
Late Collisions		UDINT	N/A		パケット伝送で512ビット目より後でコリジョンが検出された回数。
Excessive Collisions		UDINT	N/A		コリジョンが多すぎて伝送に失敗したフレーム数
MAC Transmit Errors		UDINT	N/A		MACの内部サブレイヤー受信エラーにより伝送に失敗したフレーム数
Carrier Sense Errors		UDINT	N/A		フレームの伝送試行時にキャリア検出条件が失われた、またはアサートされなかった回数
Frame Too Long		UDINT	N/A		最大許容フレームサイズを超過した受信フレーム数
MAC Receive Errors		UDINT	N/A		MACの内部サブレイヤー受信エラーにより、インターフェース上での受信に失敗したフレーム数

#	名前	アクセス	タイプ	値	コメント
6	Interface Control	Get/Set	構造体:		
	Control Bits		WORD	-	インターフェースの制御ビット
	Forced Interface Speed		UINT	-	インターフェースが動作を強制される速度。オートネゴシエーションが有効の場合は 'Object state Conflict' を返します
7	Interface Type	Get	USINT	-	下表「Interface State」を参照してください。
8	Interface State	Get	USINT	-	下表「Interface Type」を参照してください。
9	Admin State	Get/Set	USINT	-	下表「Admin State」を参照してください。
10	Interface Label	Get	SHORT_STRING	-	下表「Interface Label」を参照してください。
11	Interface Capability	Get	構造体:	-	インターフェース機能を示します
	Capability Bits		DWORD	-	speed/duplex以外のインターフェース機能 下表「Interface Capability」を参照してください。
	Speed/Duplex Options		構造体:	-	Interface Controlアトリビュートでサポートされているspeed/duplexのペアを示します
			USINT	-	speed/duplexの配列数
			構造体の配列:	-	speed/duplex配列
			UINT	-	インターフェースの速度
			USINT	-	インターフェースの二重通信モード 0 = 半二重 1 = 全二重 2 ~ 255 = 予約

- アトリビュート#6のサポートは、EtherNet/IPホストオブジェクト (F8h) にアトリビュート#9を実装することによって無効にできます ([EtherNet/IPホストオブジェクト \(F8h\)](#), ページ 175を参照)。
- アトリビュート#9のサポートは、イーサネットホストオブジェクト (F9h) にポート状態アトリビュート (#12または#13) を実装することによって無効にできます ([Ethernetホストオブジェクト \(F9h\)](#), ページ 184を参照)。

Interface Flags

ビット	名前	説明
0	Link status	イーサネット802.3通信インターフェースが、稼働しているネットワークと接続しているか否かを示します。 値: 意味: 0 非アクティブリンク 1 アクティブリンク
1	Half/full duplex	現在使用中のduplexを示します。 値: 意味: 0 半二重 1 全二重
2 - 4	Negotiation Status	リンクオートネゴシエーションの状態を示します。 値: 意味: 0 オートネゴシエーション進行中。 1 オートネゴシエーションとスピード検出に失敗 (デフォルト値を使用) (推奨されるデフォルト値は 10 Mbps、半二重) 2 オートネゴシエーション失敗、スピード検出成功 (デフォルトのduplex値を使用) 3 スピードとduplexのネゴシエーションに成功。 4 オートネゴシエーション未試行。スピードとduplexが強制された。
5	Manual Setting requires Reset	値: 意味: 0 インターフェースはランタイム中にリンクパラメータの変更を有効にできません 1 変更を適用するにはリセットが必要です
6	Local Hardware Fault	値: 意味: 0 ローカルハードウェアの障害が検出されませんでした 1 ローカルハードウェアの障害が検出されました
7-31	(予約)	0に設定。

Interface State

このアトリビュートはインターフェースの現在の運用状態を表します。

値	説明
0	不明なインターフェースの状態。
1	インターフェースは有効であり、データ送受信可能。
2	インターフェースは無効。
3	インターフェースはテスト中。

Admin State

このアトリビュートはインターフェース状態のうち管理設定を制御します。

値	説明
0	(予約)
1	インターフェースを有効化。
2	インターフェースを無効化。
3-255	(予約)

Interface Label

このアトリビュートはEtherNet/IPホストオブジェクトを介して設定できます (175ページを参照)。

インスタンス	値
1	ポート1
2	ポート2
3	内部

Interface Type

インスタンス	値	説明
1	2	ツイストペア
2	2	ツイストペア
3	1	内部インターフェース

Interface Capability

ビット	名前	説明	実装
0	Manual setting requires reset	Interface Controlアトリビュート (#6) に対して行われた変更を適用するにはデバイスのリセットが必要か否かを示します。 0 デバイスはInterface Controlアトリビュート (#6) に対して行われた変更を自動的に適用するため、変更適用のためのリセットは不要であることを示します。このビットは、Interface Controlアトリビュート (#6) が実装されていない場合、この値になります。 1 1 = デバイスはInterface Controlアトリビュート (#6) に対して行われた変更を自動的に適用しないため、変更を適用するにはリセットが必要であることを示します。 注: 以前のオブジェクトリビジョンとの後方互換性を維持するため、このビットはInterface Flagアトリビュート (#2) においても複製されます。	戻り値0
1	Auto-negotiate	0 インターフェースがオートネゴシエーションのリンクをサポートしないことを示します 1 インターフェースがオートネゴシエーションのリンクをサポートすることを示します	0は内部インターフェース、1は外部インターフェース
2	Auto-MDIX	0 インターフェースがauto MDIX動作をサポートしないことを示します 1 インターフェースがauto MDIX動作をサポートすることを示します	0は内部インターフェース、1は外部インターフェース
3	Manual speed/duplex	0 インターフェースがspeed/duplexの手动設定をサポートしないことを示します。Interface Controlアトリビュート (#6) はサポートされません。 1 インターフェースがInterface Controlアトリビュート (#6) を介してspeed/duplexの手动設定をサポートすることを示します	0は内部インターフェース、1は外部インターフェース
4 - 31	予約	0に設定されます	戻り値0

11 Anybus モジュールオブジェクト

11.1 基本情報

この章では、Anybusモジュールオブジェクトの実装とそれらがAnybus CompactCom 40 EtherNet/IPの機能とどの様に対応するかについて詳しく述べています。

標準オブジェクト:

- [Anybusオブジェクト \(01h\), ページ 115](#)
- [診断オブジェクト \(02h\), ページ 116](#)
- [Network オブジェクト \(03h\), ページ 117](#)
- [ネットワークコンフィグレーションオブジェクト \(04h\), ページ 118](#)

ネットワーク固有のオブジェクト:

- [ソケットインターフェースオブジェクト \(07h\), ページ 127](#)
- [SMTPクライアントオブジェクト \(09h\), ページ 144](#)
- [Anybusファイルシステムインターフェースオブジェクト \(0Ah\), ページ 149](#)
- [ネットワークイーサネットオブジェクト \(0Ch\), ページ 150](#)
- [CIPポートコンフィグレーションオブジェクト \(0Dh\), ページ 152](#)
- [ファンクショナルセーフティモジュール・オブジェクト \(11h\), ページ 154](#)

11.2 Anybusオブジェクト (01h)

カテゴリ

基本

オブジェクトの説明

このオブジェクトは、すべての一般的なAnybusデータをアセンブルします。完全な詳細は全般的なAnybus CompactCom 40 Software Design Guideに記載されています。

サポートされているコマンド

オブジェクト : Get_Attribute
 インスタンス : Get_Attribute
 Set_Attribute
 Get_Enum_String

オブジェクトアトリビュート (インスタンス#0)

(詳細については、全般的な Anybus CompactCom 40 Software Design Guideを参照してください。)

インスタンスアトリビュート (インスタンス#1)

基本

#	名前	アクセス	タイプ	値
1	Module type	Get	UINT16	0403h (Standard Anybus CompactCom 40)
2...11	-	-	-	詳細については、全般的な『Anybus CompactCom 40 Software Design Guide』をご参照ください。
12	LED colors	Get	構造体: UINT8 (LED1A) UINT8 (LED1B) UINT8 (LED2A) UINT8 (LED2B)	値: 色: 01h 緑 02h 赤 01h 緑 02h 赤
13...16	-	-	-	詳細については、全般的な『Anybus CompactCom 40 Software Design Guide』をご参照ください。

拡張

#	名前	アクセス	タイプ	値
17	Virtual attributes	Get/Set	-	詳細については全般的な『Anybus CompactCom 40 Software Design Guide』を参照してください。
18	Black list/White list	Get/Set		
19	Network time	Get	UINT64	CIP syncが有効になっていてデバイスがPTPマスターに同期されている場合、ネットワーク時間はIEEE-1588に基づく64ビット値としてここに表示されます。「エポック 23:59:51.51.999918, December 31, 1969」をベースとする64ビットのナノ秒カウンタとして表されます。ゼロから始めて、同期されるまでカウントします。CIP syncが無効になっている場合は、値0が返されます。

11.3 診断オブジェクト (02h)

カテゴリ

基本

オブジェクトの説明

このオブジェクトは、ホストアプリケーションのイベントと診断を処理するための、標準化された方法を提供します。

完全な詳細は全般的なAnybus CompactCom 40 Software Design Guideに記載されています。

サポートされているコマンド

オブジェクト : Get_Attribute

Create

Delete

インスタンス : Get_Attribute

オブジェクトアトリビュート (インスタンス#0)

#	名前	アクセス	データ型	値
1~4	-	-	-	詳細については、全般的な『Anybus CompactCom 40 Software Design Guide』をご参照ください。
11	Max no. of instances	Get	UINT16	5+1 (最大インスタンス数の中でも常に1つのインスタンスは、モジュールを 'EXCEPTION' 状態に強制するために、'Major, unrecoverble (重大、復旧不可)' の重大度レベルのイベント用に予約しておかなければなりません。)
12	Supported functionality	Get	BITS32	ビット0: 「0」 (イベントのラッチのサポートなし) ビット1~31: 予約 (「0」にすること)

インスタンスアトリビュート (インスタンス#1)

拡張

#	名前	アクセス	データ型	値
1	Severity	Get	UINT8	詳細については全般的な『Anybus CompactCom 40 Software Design Guide』を参照してください。
2	Event Code	Get	UINT8	
3	-	-	-	製品に未実装
4	Slot	Get	UINT16	詳細については全般的な『Anybus CompactCom 40 Software Design Guide』を参照してください。
5	ADI	Get	UINT16	
6	Element	Get	UINT8	
7	ビット	Get	UINT8	

アトリビュート#2と#4~7はネットワークに表すことはできず、モジュールによって無視されます。

この実装ではすべてのインスタンスの重大度レベルが組み合わせられ (論理ORを使用)、CIP Identity オブジェクトによってネットワーク上で表されます。

11.4 Network オブジェクト (03h)

カテゴリ

基本

オブジェクトの説明

このオブジェクトの詳細については、全般的な *Anybus CompactCom 40 Software Design Guide* を参照してください。

サポートされているコマンド

オブジェクト : Get_Attribute

インスタンス : Get_Attribute

 Set_Attribute

 Get_Enum_String

 Map_ADI_Write_Area

 Map_ADI_Read_Area

 Map_ADI_Write_Ext_Area

 Map_ADI_Read_Ext_Area

オブジェクトアトリビュート (インスタンス#0)

(詳細については、全般的な *Anybus CompactCom 40 Software Design Guide* を参照してください。)

インスタンスアトリビュート (インスタンス#1)

基本

#	名前	アクセス	タイプ	値
1	Network type	Get	UINT16	009Bh (EtherNet/IP ビーコンベース、2ポート)
2	Network type string	Get	CHARの配列	「Ethernet/IP(TM)」
3	Data format	Get	ENUM	00h (LSBファースト)
4	Parameter data support	Get	BOOL	True
5	Write process data size	Get	UINT16	現在のライトプロセスデータのサイズ (バイト数) Map_ADI_Write_Areaが成功するたびに更新されます。(詳細については、全般的な <i>Anybus CompactCom 40 Software Design Guide</i> を参照してください。)
6	Read process data size	Get	UINT16	現在のリードプロセスデータのサイズ (バイト数) Map_ADI_Read_Areaが成功するたびに更新されます。(詳細については、全般的な <i>Anybus CompactCom 40 Software Design Guide</i> を参照してください。)
7	Exception Information	Get	UINT8	モジュールがEXCEPTION状態になると追加情報が得られます。 値: 意味: 00h 使用可能な情報なし 01h アセンブリ インスタンスのマッピングが無効 02h MACアドレスの欠如 (Anybus IPにのみ適用)

11.5 ネットワークコンフィグレーションオブジェクト (04h)

カテゴリ

拡張

オブジェクトの説明

このオブジェクトには、エンドユーザーが設定するネットワーク固有の設定パラメータが格納されます。このオブジェクトにリセットコマンド(工場出荷状態へのリセット)が発行されると、すべてのインスタンスがデフォルト値になります。

このオブジェクトの設定がコンフィグレーションと一致していない場合、モジュールステータスのLEDが赤く点滅してマイナーエラーがあることを示します。

IPアドレス、サブネットマスク、ゲートウェイの組み合わせが今まで使用されていた組み合わせから変化すると、モジュールは新しい設定をイーサネットホストオブジェクト (F9h) のインスタンス#1、アトリビュート#16に書き込んで、アプリケーションに通知します。

このオブジェクトについては『Anybus CompactCom 40 Software Design Guide』に詳細説明があります。

下記も参照してください。

- [通信設定, ページ 12](#)
- [TCP/IPインターフェースオブジェクト \(F5h\), ページ 106 \(CIPオブジェクト\)](#)
- [Ethernet リンクオブジェクト \(F6h\), ページ 109](#)
- [Ethernet ホストオブジェクト \(F9h\), ページ 184](#)
- [電子メールクライアント, ページ 35](#)

サポートされているコマンド

オブジェクト :	Get_Attribute
	Reset
インスタンス :	Get_Attribute
	Set_Attribute
	Get_Enum_String

オブジェクトアトリビュート (インスタンス#0)

#	名前	アクセス	データ型	値	説明
3	Number of instances	Get	UINT16	16	サポートされているインスタンス数
4	Highest instance number	Get	UINT16	20	最大のインスタンス番号

(詳細については、全般的な *Anybus CompactCom 40 Software Design Guide* を参照してください。)

インスタンスアトリビュート (インスタンス#3、IPアドレス)

モジュールリセット後に値が使用されます。

#	名前	アクセス	データ型	説明
1	Name	Get	CHARの配列	「IPアドレス」 (多言語サポート、126ページを参照)
2	Data type	Get	UINT8	04h (= UINT8)
3	Number of elements	Get	UINT8	04h (4つの要素)
4	Descriptor	Get	UINT8	07h (読み出し/書き込み/sharedアクセス)
5	値	Get/Set	UINT8の配列	読み出しの場合、実際の値が受信されます。書き込みの場合、書き込まれた値はリセットされるまでアトリビュート#6に反映されます。 値の有効範囲: 0.0.0.0 ~ 255.255.255.255 (デフォルト =0.0.0.0)
6	Configured Value	Get	UINT8の配列	設定された値が格納されます。この値はモジュールのリセット後にアトリビュート#5に書き込まれます。 値の有効範囲: 0.0.0.0 ~ 255.255.255.255 (デフォルト =0.0.0.0)

インスタンスアトリビュート (インスタンス#4、サブネットマスク)

モジュールリセット後に値が使用されます。

#	名前	アクセス	データ型	説明
1	Name	Get	CHARの配列	「サブネットマスク」 (多言語サポート、126ページを参照)
2	Data type	Get	UINT8	04h (= UINT8)
3	Number of elements	Get	UINT8	04h (4つの要素)
4	Descriptor	Get	UINT8	07h (読み出し/書き込み/sharedアクセス)
5	値	Get/Set	UINT8の配列	読み出しの場合、実際の値が受信されます。書き込みの場合、書き込まれた値はリセットされるまでアトリビュート#6に反映されます。 値の有効範囲: 0.0.0.0 ~ 255.255.255.255 (デフォルト =0.0.0.0)
6	Configured Value	Get	UINT8の配列	設定された値が格納されます。この値はモジュールのリセット後にアトリビュート#5に書き込まれます。 値の有効範囲: 0.0.0.0 ~ 255.255.255.255 (デフォルト =0.0.0.0)

インスタンスアトリビュート (インスタンス#5、ゲートウェイアドレス)

モジュールリセット後に値が使用されます。

#	名前	アクセス	データ型	説明
1	Name	Get	CHARの配列	「ゲートウェイ」 (多言語サポート、126ページを参照)
2	Data type	Get	UINT8	04h (= UINT8)
3	Number of elements	Get	UINT8	04h (4つの要素)
4	Descriptor	Get	UINT8	07h (読み出し/書き込み/sharedアクセス)
5	値	Get/Set	UINT8の配列	読み出しの場合、実際の値が受信されます。書き込みの場合、書き込まれた値はリセットされるまでアトリビュート#6に反映されます。 値の有効範囲: 0.0.0.0 ~ 255.255.255.255 (デフォルト =0.0.0.0)
6	Configured Value	Get	UINT8の配列	設定された値が格納されます。この値はモジュールのリセット後にアトリビュート#5に書き込まれます。 値の有効範囲: 0.0.0.0 ~ 255.255.255.255 (デフォルト =0.0.0.0)

インスタンスアトリビュート (インスタンス#6、DHCP Enable)

モジュールリセット後に値が使用されます。

#	名前	アクセス	データ型	説明									
1	Name	Get	CHARの配列	「DHCP」 (多言語サポート、126ページを参照)									
2	Data type	Get	UINT8	08h (= ENUM)									
3	Number of elements	Get	UINT8	01h (1要素)									
4	Descriptor	Get	UINT8	07h (読み出し/書き込み/sharedアクセス)									
5	値	Get/Set	ENUM	読み出しの場合、実際の値が受信されます。書き込みの場合、書き込まれた値はリセットされるまでアトリビュート#6に反映されます。 (多言語サポート、126ページを参照) <table border="1"> <thead> <tr> <th>値</th> <th>文字列</th> <th>意味</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>「Disable」</td> <td>DHCP無効</td> </tr> <tr> <td>01h</td> <td>「Enable」</td> <td>DHCP有効 (デフォルト)</td> </tr> </tbody> </table>	値	文字列	意味	00h	「Disable」	DHCP無効	01h	「Enable」	DHCP有効 (デフォルト)
値	文字列	意味											
00h	「Disable」	DHCP無効											
01h	「Enable」	DHCP有効 (デフォルト)											
6	Configured Value	Get	ENUM	設定された値が格納されます。この値はモジュールのリセット後にアトリビュート#5に書き込まれます。 <table border="1"> <thead> <tr> <th>値</th> <th>文字列</th> <th>意味</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>「Disable」</td> <td>DHCP無効</td> </tr> <tr> <td>01h</td> <td>「Enable」</td> <td>DHCP有効</td> </tr> </tbody> </table>	値	文字列	意味	00h	「Disable」	DHCP無効	01h	「Enable」	DHCP有効
値	文字列	意味											
00h	「Disable」	DHCP無効											
01h	「Enable」	DHCP有効											

インスタンスアトリビュート (インスタンス#7、Ethernet 通信設定1)

変更は直ちに有効になります。

#	名前	アクセス	データ型	説明																		
1	Name	Get	CHARの配列	「Comm 1」 (多言語サポート、126ページを参照)																		
2	Data type	Get	UINT8	08h (= ENUM)																		
3	Number of elements	Get	UINT8	01h (1要素)																		
4	Descriptor	Get	UINT8	07h (読み出し/書き込み/sharedアクセス)																		
5	値	Get/Set	ENUM	値 <table border="1"> <thead> <tr> <th>値</th> <th>文字列</th> <th>意味</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>「Auto」</td> <td>オートネゴシエーション (デフォルト)</td> </tr> <tr> <td>01h</td> <td>「10 HDX」</td> <td>10Mbit、半二重</td> </tr> <tr> <td>02h</td> <td>「10 FX」</td> <td>10Mbit、全二重</td> </tr> <tr> <td>03h</td> <td>「100HDX」</td> <td>100Mbit、半二重</td> </tr> <tr> <td>04h</td> <td>「100FX」</td> <td>100Mbit、全二重</td> </tr> </tbody> </table> (多言語サポート、126ページを参照)	値	文字列	意味	00h	「Auto」	オートネゴシエーション (デフォルト)	01h	「10 HDX」	10Mbit、半二重	02h	「10 FX」	10Mbit、全二重	03h	「100HDX」	100Mbit、半二重	04h	「100FX」	100Mbit、全二重
値	文字列	意味																				
00h	「Auto」	オートネゴシエーション (デフォルト)																				
01h	「10 HDX」	10Mbit、半二重																				
02h	「10 FX」	10Mbit、全二重																				
03h	「100HDX」	100Mbit、半二重																				
04h	「100FX」	100Mbit、全二重																				
6	Configured Value	Get	ENUM	設定された値が格納されます。この値はモジュールのリセット後にアトリビュート#5に書き込まれます。 <table border="1"> <thead> <tr> <th>値</th> <th>文字列</th> <th>意味</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>「Auto」</td> <td>オートネゴシエーション</td> </tr> <tr> <td>01h</td> <td>「10 HDX」</td> <td>10Mbit、半二重</td> </tr> <tr> <td>02h</td> <td>「10 FX」</td> <td>10Mbit、全二重</td> </tr> <tr> <td>03h</td> <td>「100HDX」</td> <td>100Mbit、半二重</td> </tr> <tr> <td>04h</td> <td>「100FX」</td> <td>100Mbit、全二重</td> </tr> </tbody> </table> (多言語サポート、126ページを参照)	値	文字列	意味	00h	「Auto」	オートネゴシエーション	01h	「10 HDX」	10Mbit、半二重	02h	「10 FX」	10Mbit、全二重	03h	「100HDX」	100Mbit、半二重	04h	「100FX」	100Mbit、全二重
値	文字列	意味																				
00h	「Auto」	オートネゴシエーション																				
01h	「10 HDX」	10Mbit、半二重																				
02h	「10 FX」	10Mbit、全二重																				
03h	「100HDX」	100Mbit、半二重																				
04h	「100FX」	100Mbit、全二重																				

インスタンスアトリビュート (インスタンス#8、Ethernet 通信設定2)

変更は直ちに有効になります。

#	名前	アクセス	データ型	説明																					
1	Name	Get	CHARの配列	「Comm 2」 (多言語サポート、 126 ページを参照)																					
2	Data type	Get	UINT8	08h (= ENUM)																					
3	Number of elements	Get	UINT8	01h (1要素)																					
4	Descriptor	Get	UINT8	07h (読み出し/書き込み/sharedアクセス)																					
5	値	Get/Set	ENUM	<table border="1"> <thead> <tr> <th>値</th> <th>文字列</th> <th>意味</th> </tr> </thead> <tbody> <tr> <td colspan="3">(多言語サポート、126ページを参照)</td> </tr> <tr> <td>00h</td> <td>「Auto」</td> <td>オートネゴシエーション (デフォルト)</td> </tr> <tr> <td>01h</td> <td>「10 HDX」</td> <td>10Mbit、半二重</td> </tr> <tr> <td>02h</td> <td>「10 FX」</td> <td>10Mbit、全二重</td> </tr> <tr> <td>03h</td> <td>「100HDX」</td> <td>100Mbit、半二重</td> </tr> <tr> <td>04h</td> <td>「100FX」</td> <td>100Mbit、全二重</td> </tr> </tbody> </table>	値	文字列	意味	(多言語サポート、 126 ページを参照)			00h	「Auto」	オートネゴシエーション (デフォルト)	01h	「10 HDX」	10Mbit、半二重	02h	「10 FX」	10Mbit、全二重	03h	「100HDX」	100Mbit、半二重	04h	「100FX」	100Mbit、全二重
値	文字列	意味																							
(多言語サポート、 126 ページを参照)																									
00h	「Auto」	オートネゴシエーション (デフォルト)																							
01h	「10 HDX」	10Mbit、半二重																							
02h	「10 FX」	10Mbit、全二重																							
03h	「100HDX」	100Mbit、半二重																							
04h	「100FX」	100Mbit、全二重																							
6	Configured Value	Get	ENUM	設定された値が格納されます。この値はモジュールのリセット後にアトリビュート#5に書き込まれます。 <table border="1"> <thead> <tr> <th>値</th> <th>文字列</th> <th>意味</th> </tr> </thead> <tbody> <tr> <td colspan="3">(多言語サポート、126ページを参照)</td> </tr> <tr> <td>00h</td> <td>「Auto」</td> <td>オートネゴシエーション</td> </tr> <tr> <td>01h</td> <td>「10 HDX」</td> <td>10Mbit、半二重</td> </tr> <tr> <td>02h</td> <td>「10 FX」</td> <td>10Mbit、全二重</td> </tr> <tr> <td>03h</td> <td>「100HDX」</td> <td>100Mbit、半二重</td> </tr> <tr> <td>04h</td> <td>「100FX」</td> <td>100Mbit、全二重</td> </tr> </tbody> </table>	値	文字列	意味	(多言語サポート、 126 ページを参照)			00h	「Auto」	オートネゴシエーション	01h	「10 HDX」	10Mbit、半二重	02h	「10 FX」	10Mbit、全二重	03h	「100HDX」	100Mbit、半二重	04h	「100FX」	100Mbit、全二重
値	文字列	意味																							
(多言語サポート、 126 ページを参照)																									
00h	「Auto」	オートネゴシエーション																							
01h	「10 HDX」	10Mbit、半二重																							
02h	「10 FX」	10Mbit、全二重																							
03h	「100HDX」	100Mbit、半二重																							
04h	「100FX」	100Mbit、全二重																							

インスタンスアトリビュート (インスタンス#9、DNS1)

このインスタンスはプライマリDNSサーバのアドレスを格納します。変更はリセット後に有効になります。

#	名前	アクセス	データ型	説明
1	Name	Get	CHARの配列	「DNS1」 (多言語サポート、 126 ページを参照)
2	Data type	Get	UINT8	04h (= UINT8)
3	Number of elements	Get	UINT8	04h (4つの要素)
4	Descriptor	Get	UINT8	07h (読み出し/書き込み/sharedアクセス)
5	値	Get/Set	UINT8の配列	読み出しの場合、実際の値が受信されます。書き込みの場合、書き込まれた値はリセットされるまでアトリビュート#6に反映されます。 値の有効範囲: 0.0.0.0 ~ 255.255.255.255 (デフォルト = 0.0.0.0)
6	Configured Value	Get	UINT8の配列	設定された値が格納されます。この値はモジュールのリセット後にアトリビュート#5に書き込まれます。 値の有効範囲: 0.0.0.0 ~ 255.255.255.255 (デフォルト = 0.0.0.0)

インスタンスアトリビュート (インスタンス#10、DNS2)

このインスタンスはセカンダリDNSサーバのアドレスを格納します。変更はリセット後に有効になります。

#	名前	アクセス	データ型	説明
1	Name	Get	CHARの配列	「DNS2」 (多言語サポート、 126 ページを参照)
2	Data type	Get	UINT8	04h (= UINT8)
3	Number of elements	Get	UINT8	04h (4つの要素)
4	Descriptor	Get	UINT8	07h (読み出し/書き込み/sharedアクセス)
5	値	Get/Set	UINT8の配列	読み出しの場合、実際の値が受信されます。書き込みの場合、書き込まれた値はリセットされるまでアトリビュート#6に反映されます。 値の有効範囲: 0.0.0.0 ~ 255.255.255.255 (デフォルト =0.0.0.0)
6	Configured Value	Get	UINT8の配列	設定された値が格納されます。この値はモジュールのリセット後にアトリビュート#5に書き込まれます。 値の有効範囲: 0.0.0.0 ~ 255.255.255.255 (デフォルト =0.0.0.0)

インスタンスアトリビュート (インスタンス#11、ホスト名)

このインスタンスはモジュールのホスト名を格納します。変更はリセット後に有効になります。

#	名前	アクセス	データ型	説明
1	Name	Get	CHARの配列	「ホスト名」 (多言語サポート、 126 ページを参照)
2	Data type	Get	UINT8	07h (= CHAR)
3	Number of elements	Get	UINT8	40h (64個の要素)
4	Descriptor	Get	UINT8	07h (読み出し/書き込み/sharedアクセス)
5	値	Get/Set	CHARの配列	読み出しの場合、実際の値が受信されます。書き込みの場合、書き込まれた値はリセットされるまでアトリビュート#6に反映されます。 ホスト名、64文字
6	Configured Value	Get	CHARの配列	設定された値が格納されます。この値はモジュールのリセット後にアトリビュート#5に書き込まれます。 ホスト名、64文字

インスタンスアトリビュート (インスタンス#12、ドメイン名)

このインスタンスはドメイン名を格納します。変更はリセット後に有効になります。

#	名前	アクセス	データ型	説明
1	Name	Get	CHARの配列	「ホスト名」 (多言語サポート、 126 ページを参照)
2	Data type	Get	UINT8	07h (= CHAR)
3	Number of elements	Get	UINT8	30h (48個の要素)
4	Descriptor	Get	UINT8	07h (読み出し/書き込み/sharedアクセス)
5	値	Get/Set	CHARの配列	読み出しの場合、実際の値が受信されます。書き込みの場合、書き込まれた値はリセットされるまでアトリビュート#6に反映されます。 ドメイン名、48文字
6	Configured Value	Get	CHARの配列	設定された値が格納されます。この値はモジュールのリセット後にアトリビュート#5に書き込まれます。 ドメイン名、48文字

インスタンスアトリビュート (インスタンス#13、SMTPサーバ)

このインスタンスはSMTPサーバアドレスを格納します。変更はリセット後に有効になります。

#	名前	アクセス	データ型	説明
1	Name	Get	CHARの配列	「SMTPサーバ」 (多言語サポート、126ページを参照)
2	Data type	Get	UINT8	07h (= CHAR)
3	Number of elements	Get	UINT8	40h (64個の要素)
4	Descriptor	Get	UINT8	07h (読み出し/書き込み/sharedアクセス)
5	値	Get/Set	CHARの配列	読み出しの場合、実際の値が受信されます。書き込みの場合、書き込まれた値はリセットされるまでアトリビュート#6に反映されます。 SMTPサーバアドレス、64文字。
6	Configured Value	Get	CHARの配列	設定された値が格納されます。この値はモジュールのリセット後にアトリビュート#5に書き込まれます。 SMTPサーバアドレス、64文字。

インスタンスアトリビュート (インスタンス#14、SMTPユーザー)

このインスタンスはSMTPアカウントのユーザー名を格納します。変更はリセット後に有効になります。

#	名前	アクセス	データ型	説明
1	Name	Get	CHARの配列	「SMTPユーザー」 (多言語サポート、126ページを参照)
2	Data type	Get	UINT8	07h (= CHAR)
3	Number of elements	Get	UINT8	40h (64個の要素)
4	Descriptor	Get	UINT8	07h (読み出し/書き込み/sharedアクセス)
5	値	Get/Set	CHARの配列	読み出しの場合、実際の値が受信されます。書き込みの場合、書き込まれた値はリセットされるまでアトリビュート#6に反映されます。 SMTPアカウントユーザー名、64文字
6	Configured Value	Get	CHARの配列	設定された値が格納されます。この値はモジュールのリセット後にアトリビュート#5に書き込まれます。 SMTPアカウントユーザー名、64文字

インスタンスアトリビュート (インスタンス#15、SMTPパスワード)

このインスタンスはSMTPアカウントのパスワードを格納します。変更はリセット後に有効になります。

#	名前	アクセス	データ型	説明
1	Name	Get	CHARの配列	「SMTP Pswd」 (多言語サポート、126ページを参照)
2	Data type	Get	UINT8	07h (= CHAR)
3	Number of elements	Get	UINT8	40h (64個の要素)
4	Descriptor	Get	UINT8	07h (読み出し/書き込み/sharedアクセス)
5	値	Get/Set	CHARの配列	読み出しの場合、実際の値が受信されます。書き込みの場合、書き込まれた値はリセットされるまでアトリビュート#6に反映されます。 SMTPアカウントパスワード、64文字
6	Configured Value	Get	CHARの配列	設定された値が格納されます。この値はモジュールのリセット後にアトリビュート#5に書き込まれます。 SMTPアカウントパスワード、64文字

インスタンスアトリビュート (インスタンス#16、MDI 1設定)

このインスタンスはMDI/MDIX 1の設定を格納します。変更は直ちに有効になります。

#	名前	アクセス	データ型	説明
1	Name	Get	CHARの配列	「MDI 1」
2	Data type	Get	UINT8	08h (= ENUM)
3	Number of elements	Get	UINT8	01h (1要素)
4	Descriptor	Get	UINT8	07h (読み出し/書き込み/sharedアクセス)
5	値	Get/Set	ENUM	値 (ENUM): 文字列: 意味: 00h 「Auto」 (デフォルト) 01h 「MDI」 02h 「MDIX」
5	Configured Value	Get	ENUM	設定された値が格納されます。この値はモジュールのリセット後にアトリビュート#5に書き込まれます。 値 (ENUM): 文字列: 意味: 00h 「Auto」 01h 「MDI」 02h 「MDIX」

インスタンスアトリビュート (インスタンス#17、MDI 2設定)

このインスタンスはMDI/MDIX 2の設定を格納します。変更は直ちに有効になります。

#	名前	アクセス	データ型	説明
1	Name	Get	CHARの配列	「MDI 2」
2	Data type	Get	UINT8	08h (= ENUM)
3	Number of elements	Get	UINT8	01h (1要素)
4	Descriptor	Get	UINT8	07h (読み出し/書き込み/sharedアクセス)
5	値	Get/Set	ENUM	値 (ENUM): 文字列: 意味: 00h 「Auto」 (デフォルト) 01h 「MDI」 02h 「MDIX」
5	Configured Value	Get	ENUM	設定された値が格納されます。この値はモジュールのリセット後にアトリビュート#5に書き込まれます。 値 (ENUM): 文字列: 意味: 00h 「Auto」 01h 「MDI」 02h 「MDIX」

インスタンスアトリビュート (インスタンス#18 ~ 19)

これらのインスタンスは今後のアトリビュート用に予約されています。

インスタンスアトリビュート (インスタンス#20、QuickConnect)

このインスタンスはアプリケーションからQuickConnect機能を有効/無効にします。変更はリセット後またはパワーサイクル (電源オフオン) 後に適用されます。TCP/IPインターフェースオブジェクト (F5h) にあるQuickConnectアトリビュート (#12) の値は直ちに変わります。

このインスタンスは、EtherNet/IPホストオブジェクトでQuickConnectが有効になっていない限り、効果はありません。EtherNet/IPホストオブジェクトでQuickConnectが無効になっている場合は、エンドユーザーに対してこのインスタンスをアプリケーションで非表示にすることを推奨します。

下記も参照してください。

- [TCP/IPインターフェースオブジェクト \(F5h\), ページ 106](#)
- [EtherNet/IPホストオブジェクト \(F8h\), ページ 175](#)

#	名前	アクセス	データ型	説明				
1	Name	Get	CHARの配列	「QuickConnect」				
2	Data type	Get	UINT8	08h (= ENUM)				
3	Number of elements	Get	UINT8	01h (1要素)				
4	Descriptor	Get	UINT8	07h (読み出し/書き込み/sharedアクセス)				
5	値	Get/Set	ENUM	読み出しの場合、実際の値が受信されます。書き込みの場合、書き込まれた値はリセットされるまでアトリビュート#6に反映されます。 <table style="width: 100%; border: none;"> <tr> <td style="width: 50%;"><u>値:</u> 00h</td> <td style="width: 50%;"><u>意味:</u> 無効化 (デフォルト)</td> </tr> <tr> <td>01h</td> <td>Enable</td> </tr> </table>	<u>値:</u> 00h	<u>意味:</u> 無効化 (デフォルト)	01h	Enable
<u>値:</u> 00h	<u>意味:</u> 無効化 (デフォルト)							
01h	Enable							
6	Configured Value	Get	ENUM	設定された値が格納されます。この値はモジュールのリセット後にアトリビュート#5に書き込まれます。 <table style="width: 100%; border: none;"> <tr> <td style="width: 50%;"><u>値:</u> 00h</td> <td style="width: 50%;"><u>意味:</u> Disable</td> </tr> <tr> <td>01h</td> <td>Enable</td> </tr> </table>	<u>値:</u> 00h	<u>意味:</u> Disable	01h	Enable
<u>値:</u> 00h	<u>意味:</u> Disable							
01h	Enable							

多言語文字列

このオブジェクトのインスタンス名と列挙文字列は複数の言語に対応しており、現在の言語設定に応じて以下のように変換されます。

インスタンス	英語	ドイツ語	スペイン語	イタリア語	フランス語
3	IP address	IP-Adresse	Dirección IP	Indirizzo	Adresse IP
4	Subnet mask	Subnetzmaske	Masac. subred	Sottorete	Sous-réseau
5	Gateway	Gateway	Pasarela	Gateway	Passerelle
6	DHCP	DHCP	DHCP	DHCP	DHCP
	Enable	Einschalten	Activado	Abilitato	Activé
	Disable	Ausschalten	Desactivado	Disabilitato	Désactivé
7	Comm 1	Komm 1	Comu 1	Connessione 1	Comm 1
	Auto	Auto	Auto	Auto	Auto
	10 HDX	10 HDX	10 HDX	10 HDX	10 HDX
	10 FDX	10 FDX	10 FDX	10 FDX	10 FDX
	100 HDX	100 HDX	100 HDX	100 HDX	100 HDX
	100 FDX	100FDX	100 FDX	100 FDX	100 FDX
8	Comm 2	Komm 2	Comu 2	Connessione 2	Comm 2
	Auto	Auto	Auto	Auto	Auto
	10 HDX	10 HDX	10 HDX	10 HDX	10 HDX
	10 FDX	10 FDX	10 FDX	10 FDX	10 FDX
	100 HDX	100 HDX	100 HDX	100 HDX	100 HDX
100 FDX	100FDX	100 FDX	100 FDX	100 FDX	
9	DNS1	DNS 1	DNS Primaria	DNS1	DNS1
10	DNS2	DNS 2	DNS Secundia.	DNS2	DNS2
11	Host name	Host name	Nombre Host	Nome Host	Nom hôte
12	Domain name	Domain name	Nobre Domain	Nome Dominio	Dom Domaine
13	SMTP Server	SMTP Server	Servidor SMTP	Server SMTP	SMTP serveur
14	SMTP User	SMTP User	Usuario SMTP	Utente SMTP	SMTP utiliza.
15	SMTP Pswd	SMTP PSWD	Clave SMTP	Password SMTP	SMTP mt passe

11.6 ソケットインターフェースオブジェクト (07h)

カテゴリ

拡張

オブジェクトの説明

このオブジェクトでTCP/IPスタックソケットインターフェースに直接アクセスでき、カスタム プロトコルをTCP/UDPの上を実装することを可能にします。

このオブジェクトにアクセスするために使用されるコマンドの一部には、分割が必要な場合があります。メッセージは、送信または受信データ量がメッセージチャネルが扱える大きさよりも大きい場合、分割されます。詳細については、[メッセージの分割](#)、[ページ 142](#)を参照してください。



このオブジェクトにより提供される機能はソケットインターフェースプログラミングに習熟しており、TCP/IPプログラミングに関わるコンセプトを完璧に理解したユーザーのみが試みることができます。

サポートされているコマンド

オブジェクト :	Get_Attribute
	Create (下記を参照)
	Delete (下記を参照)
	DNS_Lookup (下記を参照)
インスタンス :	Get_Attribute
	Set_Attribute
	Bind (下記を参照)
	Shutdown (下記を参照)
	Listen (下記を参照)
	Accept (下記を参照)
	Connect (下記を参照)
	Receive (下記を参照)
	Receive_From (下記を参照)
	Send (下記を参照)
	Send_To (下記を参照)
	P_Add_membership (下記を参照)
	IP_Drop_membership (下記を参照)

オブジェクトアトリビュート (インスタンス#0)

#	名前	アクセス	データ型	値
1	Name	Get	CHARの配列	「Socket interface」
2	Revision	Get	UINT8	01h
3	Number of instances	Get	UINT16	オープンしたソケット数
4	Highest instance no.	Get	UINT16	作成されたインスタンスの番号のうち最も大きい番号
11	Max. no. of instances	Get	UINT16	0008h (8個のインスタンス) : BACnet/IP 0014h (20個のインスタンス) : その他すべての産業用イーサネット

インスタンスアトリビュート (ソケット#1 ~ Max. no. of instances)

拡張

#	名前	アクセス	データ型	説明
1	Socket Type	Get	UINT8	<p>値: Socket Type</p> <p>00h SOCK_STREAM, NONBLOCKING (TCP)</p> <p>01h SOCK_STREAM, BLOCKING (TCP)</p> <p>02h SOCK_DGRAM, NONBLOCKING (UDP)</p> <p>03h SOCK_DGRAM, BLOCKING (UDP)</p>
2	Port	Get	UINT16	ソケットが割り当てられているローカルポート
3	Host IP	Get	UINT32	Host IPアドレスまたは未接続なら0 (ゼロ)
4	Host port	Get	UINT16	ホストのポート番号、未接続なら0 (ゼロ)
5	TCP State	Get	UINT8	<p>State (TCPソケットのみ):</p> <p>値 ステート/説明</p> <p>00h CLOSED 閉じた</p> <p>01h LISTEN 接続待機中</p> <p>02h SYN_SENT アクティブ、SYNを送受信済み</p> <p>03h SYN_RECEIVED SYNを送受信済み</p> <p>04h ESTABLISHED 確立済み。</p> <p>05h CLOSE_WAIT FINを受信して、クローズ待ち</p> <p>06h FIN_WAIT_1 クローズして、FINを送信済み</p> <p>07h CLOSING 交換済みFINを閉じて、FIN ACK待ち</p> <p>08h LAST_ACK FINを送信、クローズして、FIN ACK待ち</p> <p>09h FIN_WAIT_2 クローズして、FINが確認された</p> <p>Ah TIME_WAIT クローズ後静かに待つ</p>
6	TCP RX bytes	Get	UINT16	RXバッファーにあるバイト数 (TCPソケットのみ)
7	TCP TX bytes	Get	UINT16	TXバッファーのバイト数 (TCPソケットのみ)
8	Reuse address	Get/Set	BOOL	<p>ソケットはローカルアドレスを再使用可能</p> <p>値 意味</p> <p>1 有効</p> <p>0 無効 (デフォルト)</p>
9	Keep alive	Get/Set	BOOL	<p>プロトコルがアイドルコネクションを探します (TCPソケットのみ)。</p> <p>Keep alive アトリビュートが設定されていれば、コネクションは120分アイドル状態が続いた後で初めて探索されます。確認に失敗すると、接続は75秒ごとに探索され続けます。この探索が8回失敗すると接続は終了されます。</p> <p>値 意味</p> <p>1 有効</p> <p>0 無効 (デフォルト)</p>
10	IP Multicast TTL	Get/Set	UINT8	IPマルチキャストTTL値 (UDPソケットのみ)。 デフォルト = 1。
11	IP Multicast Loop	Get/Set	BOOL	<p>IPマルチキャストループバック (UDPソケットのみ)</p> <p>ループバックメッセージを取得するためにはグループに属していなければなりません</p> <p>値 意味</p> <p>1 有効 (デフォルト)</p> <p>0 無効</p>
12	(予約)			
13	TCP No Delay	Get/Set	BOOL	<p>合体パケットへの送出手を遅延させない (TCP)。</p> <p>値 意味</p> <p>1 遅延 (デフォルト)</p> <p>0 遅延しない (ソケットのNagleアルゴリズムを無効化する)</p>
14	TCP Connect Timeout	Get/Set	UINT16	TCP接続タイムアウト、秒数 (デフォルト = 75s)

コマンドの詳細: Create

カテゴリ

拡張

詳細

コマンドコード	03h
有効な対象:	オブジェクトインスタンス

説明

このコマンドはソケットを作成します。

このコマンドは WAIT_PROCESS、IDLE、PROCESS_ACTIVE の各ステートでのみ使用できます。

• コマンド詳細

フィールド	内容										
CmdExt[0]	(予約、ゼロに設定)										
CmdExt[1]	<table> <thead> <tr> <th>値:</th> <th>ソケットタイプ:</th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>SOCK_STREAM, NON-BLOCKING (TCP)</td> </tr> <tr> <td>01h</td> <td>SOCK_STREAM, BLOCKING (TCP)</td> </tr> <tr> <td>02h</td> <td>SOCK_DGRAM, NON-BLOCKING (UDP)</td> </tr> <tr> <td>03h</td> <td>SOCK_DGRAM, BLOCKING (UDP)</td> </tr> </tbody> </table>	値:	ソケットタイプ:	00h	SOCK_STREAM, NON-BLOCKING (TCP)	01h	SOCK_STREAM, BLOCKING (TCP)	02h	SOCK_DGRAM, NON-BLOCKING (UDP)	03h	SOCK_DGRAM, BLOCKING (UDP)
値:	ソケットタイプ:										
00h	SOCK_STREAM, NON-BLOCKING (TCP)										
01h	SOCK_STREAM, BLOCKING (TCP)										
02h	SOCK_DGRAM, NON-BLOCKING (UDP)										
03h	SOCK_DGRAM, BLOCKING (UDP)										

• 応答詳細

フィールド	内容	コメント
Data[0]	インスタンス番号 (下位)	作成されたソケットのインスタンス番号。
Data[1]	インスタンス番号 (上位)	

コマンドの詳細: Delete

カテゴリ

拡張

詳細

コマンドコード 04h

有効な対象 : オブジェクトインスタンス

説明

このコマンドは以前に作成されたソケットを削除し、(接続されていれば) コネクションをクローズします。

- ソケットがTCPタイプであり、コネクションが確立されていれば、接続はRSTフラグによって終了されます。
- TCPコネクションを正常に終了したい場合は、コネクションが代わりにFINフラグで終了されるように、ソケットを削除する前に 'Shutdown' コマンド (下記を参照) を使用する事を推奨します。

- コマンド詳細

フィールド	内容	コメント
CmdExt[0]	削除するインスタンス番号 (下位)	削除されるソケットのインスタンス番号。
CmdExt[1]	削除するインスタンス番号 (上位)	

- 応答詳細

(データなし)

コマンドの詳細: Bind

カテゴリ

拡張

詳細

コマンドコード 10h

有効な対象 : インスタンス

説明

このコマンドはソケットをローカルポートにバインドします。

- コマンド詳細

フィールド	内容	コメント
CmdExt[0]	要求ポート番号 (下位)	空いているポートにバインド要求をする為に0 (ゼロ) に設定。
CmdExt[1]	要求ポート番号 (上位)	

- 応答詳細

フィールド	内容	コメント
CmdExt[0]	バインドされたポート番号 (下位)	ソケットがバインドされた実際のポート。
CmdExt[1]	バインドされたポート番号 (上位)	

コマンドの詳細: Shutdown

カテゴリ

拡張

詳細

コマンドコード	11h
有効な対象:	インスタンス

説明

このコマンドはFINフラグによってTCPコネクションを終了します。この応答がコネクションが実際に切断された事を示さない事に注意して下さい。これは、このコマンドはノンブロッキングソケットのポーリングにも、ブロッキングソケットのブロックにも使用できない事を意味します。

- コマンド詳細

フィールド	内容		
CmdExt[0]	(予約、ゼロに設定)		
CmdExt[1]	<table> <tr> <td>値: 00h 01h 02h</td> <td>モード: 受信チャンネルのシャットダウン 送信チャンネルのシャットダウン 送受信チャンネルともにシャットダウン</td> </tr> </table>	値: 00h 01h 02h	モード: 受信チャンネルのシャットダウン 送信チャンネルのシャットダウン 送受信チャンネルともにシャットダウン
値: 00h 01h 02h	モード: 受信チャンネルのシャットダウン 送信チャンネルのシャットダウン 送受信チャンネルともにシャットダウン		

- 応答詳細

(データなし)

TCPコネクションを正常に切断するための推奨手順を以下に説明します。

アプリケーションからシャットダウンを開始する:

1. 01hに設定されたCmdExt[1]でシャットダウンを送信。これはFINフラグを、送信チャンネルをシャットダウンする為にホストへ送ります。受信チャンネルがまだ稼働していることに注意してください。
2. エラーメッセージオブジェクト固有エラー (EPIPE (13))が受信されるまでソケットでデータを受信します。ホストが受信チャンネルを終了したことを表します。ホストが受信チャンネルを終了しない場合、タイムアウトを使用し、手順3へ進んでください。
3. ソケットインスタンスを削除します。ステップ2がタイムアウトした場合、RSTフラグが送られてソケットを終了します。

ホストからシャットダウンを開始する:

1. ソケットでデータを受信します。ゼロバイトを受信した場合はホストがソケットの受信チャンネルをクローズしたことを表します。
2. 未送信データをホストに送るよう試みてください。
3. 01hに設定されたCmdExt[1]でシャットダウンを送信。これによりFINフラグがホストに送られて、送信チャンネルをシャットダウンします。
4. ソケットインスタンスを削除します。

コマンドの詳細: Listen

カテゴリ

拡張

詳細

コマンドコード 12h
有効な対象 : インスタンス

説明

このコマンドはTCPソケットをリスンステートに変えます。

- コマンド詳細

フィールド	内容
CmdExt[0]	(予約、ゼロに設定)
CmdExt[1]	(予約)

- 応答詳細

(データなし)

コマンドの詳細: Accept

カテゴリ

拡張

詳細

コマンドコード	13h
有効な対象:	インスタンス

説明

このコマンドはリスリングしているTCPソケットでの着信接続を受け付けます。新たなソケットインスタンスが受け付けられた各コネクションに対して作成されます。新たなソケットがホストと接続され、応答がそのインスタンス番号を返します。

NONBLOCKING mode このコマンドは着信コネクションに対して繰り返し (ポーリングされ) 発行されなくてはなりません。着信コネクションの要求が存在しなければ、モジュールはエラーコード 0006h (EWOULDBLOCK)で応答します。

BLOCKING mode このコマンドは接続要求が検出されるまでブロックします。

このコマンドは、受け付けられたコネクションに対して使用する空いたインスタンスが存在する場合のみ受け付けられます。ブロックするコネクションに対してこのコマンドはインスタンスを予約します。

- コマンド詳細
(データなし)
- 応答詳細

フィールド	内容
Data[0]	接続されたソケット用のインスタンス番号 (下位バイト)
Data[1]	接続されたソケット用のインスタンス番号 (上位バイト)
Data[2]	ホストIPアドレス バイト 4
Data[3]	ホストIPアドレス バイト 3
Data[4]	ホストIPアドレス バイト 2
Data[5]	ホストIPアドレス バイト 1
Data[6]	ホストポート番号 (下位バイト)
Data[7]	ホストポート番号 (上位バイト)

コマンドの詳細: Connect

カテゴリ

拡張

詳細

コマンドコード	14h
有効な対象:	インスタンス

説明

SOCK-DGRAMソケットに関して、このコマンドは、ソケットが関連付けられるペア(データグラムが送られるターゲットであり、データグラムが受信される唯一のアドレス)を指定します。

SOCK_STREAMソケットに関して、このコマンドはホストとの接続確立を試みます。

SOCK_STREAMソケットは一回しか接続に成功できない一方、SOCK_DGRAMソケットはこのサービスを複数回使用して関連付けを変更することがあります。SOCK-DGRAMソケットは、IPアドレス0.0.0.0、ポート0(ゼロ)と接続することによってその関連付けを解消できます。

NON-BLOCKING mode: このコマンドは接続が確立する、拒否される、またはタイムアウトするまで繰り返し(ポーリングされ)発行されなければなりません。最初の接続試行は受け入れられます。その後、このコマンドは接続を試みている間ポーリングによる要求に対してエラーコード22 (EINPROGRESS)を返します。

BLOCKING mode: このコマンドは接続が確立するか、接続要求がタイムアウトまたは接続エラーのために取り消されるまでブロックします。

- コマンド詳細

フィールド	内容
CmdExt[0]	(予約、ゼロに設定)
CmdExt[1]	
Data[0]	ホストIPアドレス バイト 4
Data[1]	ホストIPアドレス バイト 3
Data[2]	ホストIPアドレス バイト 2
Data[3]	ホストIPアドレス バイト 1
Data[4]	ホストポート番号 (下位バイト)
Data[5]	ホストポート番号 (上位バイト)

- 応答詳細

(データなし)

コマンドの詳細: Receive

カテゴリ

拡張

詳細

コマンドコード	15h
有効な対象:	インスタンス

説明

このコマンドは接続されたソケットからデータを受信します。最大1472 バイトを受信する為にメッセージ分割が使用される場合があります (詳細については、[メッセージの分割, ページ 142](#)を参照)。

SOCK_DGRAMソケットに関して、モジュールは要求されたデータ量を次に受信したデータグラムから返します。データグラムが要求されたものより小さい場合、データグラム全体が応答メッセージに含まれて返されます。データグラムが要求されたものより大きい場合、超過バイトが破棄されます。

SOCK_STREAMソケットに関しては、モジュールは受信したデータストリームから要求されたバイト数を返します。実際のデータサイズが要求分より小さければ、利用可能なすべてのデータが返されます。

NON-BLOCKING mode: ソケットにデータが無ければ、エラーコード0006h (EWOULDBLOCK) が返されます。

BLOCKING mode: モジュールは作業が終了するまでは応答を発行しません。

モジュールがデータ量0 (ゼロ) バイトで正常に応答した場合、ホストがコネクションを閉じたことを意味します。しかし送信チャンネルはまだ有効であり、**Shutdown**および**Delete**コマンドの両方またはそのいずれかで終了する必要があります。

- コマンド詳細

フィールド	内容	コメント
CmdExt[0]	(予約)	(ゼロに設定)
CmdExt[1]	分割制御ビット	詳細については、 メッセージの分割, ページ 142 を参照してください。
Data[0]	受信データサイズ (下位)	最初のセグメントでのみ使用
Data[1]	受信データサイズ (上位)	

- 応答詳細

データ応答は分割される場合があります (詳細については、[メッセージの分割, ページ 142](#)を参照)。

フィールド	内容	コメント
CmdExt[0]	(予約)	(ゼロに設定)
CmdExt[1]	分割制御ビット	詳細については、 メッセージの分割, ページ 142 を参照してください。
Data[0...n]	受信データ	-

コマンドの詳細: Receive_From

カテゴリ

拡張

詳細

コマンドコード	16h
有効な対象:	インスタンス

説明

このコマンドはデータを未接続のSOCK_DGRAMソケットから受信します。最大1472 バイトを受信する為にメッセージ分割が使用される場合があります (詳細については、[メッセージの分割, ページ 142](#)を参照)。

モジュールは要求されたデータ量を次に受信されたデータグラムから返します。データグラムが要求されたものより小さい場合、データグラム全体が応答メッセージに含まれて返されます。データグラムが要求されたものより大きい場合、超過バイトが破棄されます。

応答メッセージには送信者のIPアドレスとポート番号が含まれます。

NON-BLOCKING mode: ソケットにデータが無ければ、エラーコード0006h (EWOULDBLOCK) が返されます。

BLOCKING mode: モジュールは作業が終了するまでは応答を発行しません。

• コマンド詳細

フィールド	内容	コメント
CmdExt[0]	(予約)	(ゼロに設定)
CmdExt[1]	分割制御ビット	詳細については、 メッセージの分割, ページ 142 を参照してください。
Data[0]	受信データサイズ (下位バイト)	最初のセグメントでのみ使用
Data[1]	受信データサイズ (上位バイト)	

• 応答詳細

データ応答は分割される場合があります (詳細については、[メッセージの分割, ページ 142](#)を参照)。

フィールド	内容	コメント
CmdExt[0]	(予約)	(ゼロに設定)
CmdExt[1]	分割制御ビット	詳細については、 メッセージの分割, ページ 142 を参照してください。
Data[0]	ホストIPアドレス バイト 4	ホストアドレス/ポート情報は先頭のセグメントにのみ含まれます。その後のデータはすべてData[0]から開始されます
Data[1]	ホストIPアドレス バイト 3	
Data[2]	ホストIPアドレス バイト 2	
Data[3]	ホストIPアドレス バイト 1	
Data[4]	ホストポート番号 (下位バイト)	
Data[5]	ホストポート番号 (上位バイト)	
Data[6...n]	受信データ	

コマンドの詳細: Send

カテゴリ

拡張

詳細

コマンドコード	17h
有効な対象:	インスタンス

説明

このコマンドは接続ソケットでデータを送ります。最大1472バイトを送信する為にメッセージ分割が使用される場合があります (詳細については、[メッセージの分割, ページ 142](#)を参照)。

NON-BLOCKING mode: 送信バッファに十分なバッファースペースがない場合、モジュールはエラーコードコード0006h (EWOULDBLOCK) を返します

BLOCKING mode: 送信バッファに十分なバッファースペースがない場合、モジュールはこのスペースができるまでブロックします。

- コマンド詳細

より大きいデータ量 (すなわち、>255 バイト) を送ることができるように、コマンドデータを分割できます (詳細については、[メッセージの分割, ページ 142](#)を参照)。

フィールド	内容	コメント
CmdExt[0]	(予約)	(ゼロに設定)
CmdExt[1]	分割制御	(詳細については、 メッセージの分割, ページ 142 を参照してください。)
Data[0...n]	送信するデータ	-

- 応答詳細

フィールド	内容	コメント
CmdExt[0]	(予約)	(無視)
CmdExt[1]		
Data[0]	送信済みバイト数 (下位)	最後のセグメントでのみ有効
Data[1]	送信済みバイト数 (上位)	

コマンドの詳細: Send_To

カテゴリ

拡張

詳細

コマンドコード	18h
有効な対象 :	インスタンス

説明

このコマンドは未接続SOCK-DGRAMソケット上の指定ホストにデータを送ります。最大1472 バイトを送信する為にメッセージ分割が使用される場合があります (詳細については、Appendixおよび[メッセージの分割, ページ 142](#)を参照)。

- コマンド詳細

より大きいデータ量 (すなわち、>255 バイト) を送ることができるように、コマンドデータを分割できます (詳細については、[メッセージの分割, ページ 142](#)を参照)。

フィールド	内容	コメント
CmdExt[0]	(予約)	(ゼロに設定)
CmdExt[1]	分割制御	詳細については、 メッセージの分割, ページ 142 を参照してください。
Data[0]	ホストIPアドレス バイト 4	ホストアドレス/ポート情報は最初のセグメントにのみ含まれます。その後のデータはすべてData[0]から開始されなければなりません
Data[1]	ホストIPアドレス バイト 3	
Data[2]	ホストIPアドレス バイト 2	
Data[3]	ホストIPアドレス バイト 1	
Data[4]	ホストポート番号 (下位バイト)	
Data[5]	ホストポート番号 (上位バイト)	
Data[6...n]	送信するデータ	

- 応答詳細

フィールド	内容	コメント
CmdExt[0]	(予約)	(無視)
CmdExt[1]		
Data[0]	送信済みバイト数 (下位バイト)	最後のセグメントでのみ有効
Data[1]	送信済みバイト数 (上位バイト)	

コマンドの詳細: IP_Add_Membership

カテゴリ

拡張

詳細

コマンドコード	19h
有効な対象 :	インスタンス

説明

このコマンドはソケットをIPマルチキャストグループメンバーシップに割り当てます。モジュールは常に「すべてのホストグループ」に自動的に連結しますが、このコマンドを使用して最大20種類の追加メンバーシップを指定することができます。

- コマンド詳細

フィールド	内容
CmdExt[0]	(予約、ゼロに設定)
CmdExt[1]	
Data[0]	グループIPアドレス バイト 4
Data[1]	グループIPアドレス バイト 3
Data[2]	グループ IPアドレス バイト 2
Data[3]	グループ IPアドレス バイト 1

- 応答詳細

(データなし)

コマンドの詳細: IP_Drop_Membership

カテゴリ

拡張

詳細

コマンドコード	1Ah
有効な対象 :	インスタンス

説明

このコマンドはIPマルチキャストグループメンバーシップからソケットを削除します。

- コマンド詳細

フィールド	内容
CmdExt[0]	(予約、ゼロに設定)
CmdExt[1]	
Data[0]	グループIPアドレス バイト 4
Data[1]	グループIPアドレス バイト 3
Data[2]	グループ IPアドレス バイト 2
Data[3]	グループ IPアドレス バイト 1

- 応答詳細

(データなし)

コマンドの詳細: DNS_Lookup

カテゴリ

拡張

詳細

コマンドコード	1Bh
有効な対象 :	オブジェクト

説明

このコマンドは与えられたホスト名を解決し、IPアドレスを返します。

- コマンド詳細

フィールド	内容	コメント
CmdExt[0]	(予約)	(ゼロに設定)
CmdExt[1]		
Data[0...N]	Host name	解決するホスト名

- 応答詳細 (成功)

フィールド	内容	コメント
CmdExt[0]	(予約)	(ゼロに設定)
CmdExt[1]		
Data[0]	IPアドレス バイト 4	指定ホストのIPアドレス
Data[1]	IPアドレス バイト 3	
Data[2]	IPアドレス バイト 2	
Data[3]	IPアドレス バイト 1	

ソケットインターフェース エラーコード (オブジェクト固有)

ソケットインターフェースオブジェクトを使用するとき、以下のオブジェクト固有エラーコードがモジュールから返されることがあります。

エラーコード	Name	意味
1	ENOBUFFS	利用可能な内部バッファが無い
2	ETIMEDOUT	タイムアウトイベント発生
3	EISCONN	ソケットは接続済み
4	EOPNOTSUPP	サービス未サポート
5	ECONNABORTED	接続は取消されました
6	EWouldBlock	非ブロックソケットタイプなのでソケットのブロックは不可
7	ECONNREFUSED	接続が拒否された
8	ECONNRESET	接続リセット
9	ENOTCONN	ソケットは未接続です
10	EALREADY	ソケットはすでに要求されたモードになっています
11	EINVAL	無効なサービスデータ
12	EMSGSIZE	無効なメッセージサイズ
13	EPIPE	パイプにエラー
14	EDESTADDRREQ	ターゲットのアドレスが必要です
15	ESHUTDOWN	ソケットはすでにシャットダウンされた
16	(予約)	-
17	EHAVEOOB	帯域外のデータが利用可能
18	ENOMEM	利用できる内部メモリーがない
19	EADDRNOTAVAIL	利用可能なアドレスがない
20	EADDRINUSE	アドレスは使用中
21	(予約)	-
22	EINPROGRESS	サービスは既に実行中
28	ETOOMANYREFS	参照先が多すぎる
101	Command aborted	コマンドがソケットでブロックしておりそのソケットをDeleteコマンドで閉じると、このエラーコードがブロックしているコマンドに対して返されます。
102	DNS name error	ホスト名の解決に失敗 (DNSサーバから名前のエラー応答)。
103	DNS timeout	DNSルックアップの実行タイムアウト。
104	DNS command failed	その他のDNSエラー。

メッセージの分割

概要

カテゴリ： 拡張

Anybus CompactCom 40で可能な最大メッセージサイズは通常は1524 バイトです。一部のアプリケーションでは最大メッセージサイズが255 バイトです。例えばアプリケーションには変更を加えず、Anybus CompactCom 40でAnybus CompactCom 30を置換する場合があります。最大ソケットメッセージサイズは1472です。255 バイトを超えるソケットインターフェースメッセージに対応するため、分割プロトコルが使用されます。

i 分割ビットはすべてのソケットインターフェースメッセージに対して、メッセージの分割が必要か否かには関わらず分割を使用できるコマンドで設定されなければなりません。

この分割プロトコルはメッセージレイヤーに実装されます。シリアルホストインターフェースで使用される分割プロトコルと混同しないでください。詳細については、一般的なAnybus CompactCom 40 Software Design Guideを参照してください。

モジュールはインスタンス毎に分割が1のメッセージをサポートします。

コマンドの分割

コマンドメッセージが分割される時、コマンドイニシエーターは同じコマンドヘッダーを何回も送ります。各メッセージに対して、データフィールドは次のデータセグメントと交換されます。

コマンド分割は以下のコマンドのために使用されます (ソケットインターフェースオブジェクト固有のコマンド):

- Send
- Send To

分割コマンドを発行する際は次のルールに従ってください:

- 最初のセグメントを発行するときはFSを設定する必要があります。
- それ以降のセグメントを発行するときはFSとLSはクリアされなければなりません。
- 最後のセグメントを発行するとき、LFビットの設定が必要です。
- 単一セグメントのコマンド (サイズがメッセージチャンネルサイズ以下) についてはFSとLSの設定が必要です。
- 最後の応答メッセージは操作の実際の結果を含みます。
- コマンドを出した側はABに設定したメッセージを出すことによって随時操作を取り消すことができます。
- 分割エラーが転送中に検出された場合、エラーメッセージが返され、現在の分割メッセージは破棄されます。しかし、このことは現在のセグメントにしか適用されません。それより前に転送されたセグメントは有効です。

分割制御ビット (コマンド)

ビット	内容	意味
0	FS	現在のセグメントが最初のセグメントの場合にセットする
1	LS	現在のセグメントが最後のセグメントの場合にセットする
2	AB	分割を取り消す場合にセットする
3..7	(予約)	0(ゼロ)に設定

分割制御ビット (応答)

ビット	内容	意味
0..7	(予約)	Ignore

応答分割

応答が分割される時、コマンドを出す側は同じコマンドを繰り返し出すことで次のセグメントを要求します。各応答に対してデータフィールドは次のデータセグメントと交換されます。

応答分割は以下のコマンドへの応答のために使用されます (ソケットインターフェースオブジェクト固有のコマンド):

- Receive
- Receive From

分割された応答を受信するとき、次のルールが適用されます:

- 最初のセグメントでFSがセットされる。
- それ以降の全セグメントではFSとLSがクリアされる。
- 最後のセグメントでLSがセットされる。
- 単一セグメント応答 (サイズがメッセージチャンネルサイズ以下) についてはFSとLSがセットされます。
- コマンドを出した側はABに設定したメッセージを出すことによって随時操作を取り消すことができます。

分割制御ビット (コマンド)

ビット	内容	意味
0	(予約)	(ゼロに設定)
1		
2	AB	分割を取り消す場合にセットする
3..7	(予約)	0 (ゼロ) に設定

分割制御ビット (応答)

ビット	内容	意味
0	FS	現在のセグメントが最初のセグメントの場合にセットする
1	LS	現在のセグメントが最後のセグメントの場合にセットする
2..7	(予約)	0 (ゼロ) に設定

11.7 SMTPクライアントオブジェクト (09h)

カテゴリ

拡張

オブジェクトの説明

このオブジェクトはSMTPクライアント関連の関数をグループ化します。

サポートされているコマンド

オブジェクト :	Get_Attribute
	Create
	Delete
	Send e-mail from file (下記を参照)
インスタンス :	Get_Attribute
	Set_Attribute
	Send e-mail (下記を参照)

オブジェクトアトリビュート (インスタンス#0)

#	名前	アクセス	データ型	値
1	Name	Get	CHARの配列	「SMTP Client」
2	Revision	Get	UINT8	01h
3	Number of instances	Get	UINT16	-
4	Highest instance no.	Get	UINT16	-
11	Max. no. of instances	Get	UINT16	0006h
12	Success count	Get	UINT16	送信に成功したメッセージ数を反映する
13	Error count	Get	UINT16	届けられなかったメッセージ数を反映する

インスタンスアトリビュート (インスタンス#1)

インスタンスはアプリケーションによってダイナミックに作成されます。

#	名前	アクセス	データ型	説明
1	From	Get/Set	CHARの配列	「someone@somewhere.com」など
2	To	Get/Set	CHARの配列	「someone.else@anywhere.net」など
3	Subject	Get/Set	CHARの配列	「重要なお知らせ」など
4	Message	Get/Set	CHARの配列	「システムシャットダウン」など

コマンドの詳細: Create

カテゴリ

拡張

詳細

コマンドコード 03h
有効な対象 : オブジェクト

説明

このコマンドは電子メールインスタンスを作成します。

- コマンド詳細

フィールド	内容	コメント
CmdExt[0]	(予約)	(ゼロに設定)
CmdExt[1]		

- 応答詳細

フィールド	内容	コメント
CmdExt[0]	(予約)	(無視)
CmdExt[1]		
Data[0]	インスタンス番号	下位バイト
Data[1]		上位バイト

コマンドの詳細: Delete

カテゴリ

拡張

詳細

コマンドコード 04h
有効な対象 : オブジェクト

説明

このコマンドは電子メールインスタンスを削除します。

- コマンド詳細

フィールド	内容	コメント
CmdExt[0]	電子メールのインスタンス番号	下位バイト
CmdExt[1]		上位バイト

- 応答詳細

(データなし)

コマンドの詳細: Send E-mail From File

カテゴリ

拡張

詳細

コマンドコード 11h
有効な対象 : オブジェクト

説明

このコマンドはファイルシステムのファイルを基に電子メールを送ります。

ファイルは下記形式によるシンプルなASCIIファイルでなければなりません。

```
[To]
recipient

[From]
sender

[Subject]
email subject

[Headers]
extra headers, optional

[Message]
actual email message
```

- コマンド詳細

フィールド	内容
CmdExt[0]	(予約、ゼロに設定)
CmdExt[1]	
Data[0... n]	パス + メッセージファイルのファイル名

- 応答詳細

(データなし)

コマンドの詳細: Send E-mail

カテゴリ

拡張

詳細

コマンドコード	10h
有効な対象:	インスタンス

説明

このコマンドは指定された電子メールインスタンスを送信します。

- コマンド詳細
(データなし)
- 応答詳細
(データなし)

オブジェクト固有のエラーコード

エラーコード	意味
1	SMTPサーバが見つかりません
2	SMTPサーバの準備ができていません
3	認証エラー
4	SMTPソケットエラー
5	SSIがサポートされている場合: SSIスキャンエラー それ以外の場合は予約
6	電子メールファイルを解釈できません
255	SMTP未指定エラー
(その他)	(予約)

11.8 Anybusファイルシステムインターフェースオブジェクト (0Ah)

カテゴリ

拡張

オブジェクトの説明

このオブジェクトは内蔵ファイルシステムとのインターフェースになります。各インスタンスはファイルストリームへのハンドルを表し、ファイルシステム操作のためのサービスを含んでいます。

これによって、アプリケーション固有のWebページをインストールする際、ホストアプリケーションによるモジュールの組み込みファイルシステムへのアクセスが可能になります。

インスタンスはランタイム中に動的に作成・削除されます。

このオブジェクトについての完全な詳細は、*Anybus CompactCom 40 Software Design Guidel*に記載されています。

11.9 ネットワークイーサネットオブジェクト (0Ch)

カテゴリ

拡張

オブジェクトの説明

このオブジェクトはイーサネット固有情報をアプリケーションに提供します。

オブジェクトには3つのインスタンスがあり、それぞれ以下のポートに対応します。

インスタンス番号	ポート
1	内部ポート
2	ポート1
3	ポート2

各インスタンスは、ポートの統計カウンターを提供します。例えばこの情報は、JSONスクリプト言語を使用して、内部Webページ (存在する場合) で表示することができます。



インスタンスアトリビュート#1は予約されており、以前のアプリケーションとの後方互換性のために使用されません。

サポートされているコマンド

オブジェクト : Get_Attribute

インスタンス : Get_Attribute

オブジェクトアトリビュート (インスタンス#0)

#	名前	アクセス	データ型	値
1	Name	Get	CHARの配列	「Network Ethernet」
2	Revision	Get	UINT8	01h
3	Number of instances	Get	UINT16	3
4	Highest instance no.	Get	UINT16	3

インスタンスアトリビュート (インスタンス#1)

#	名前	アクセス	データ型	説明
1	MAC Address	Get	UINT8の配列	予約済み、後方互換性のために使用されます。 (Device MACアドレス) (Ethernet ホストオブジェクト (F9h), ページ 184も参照)
2	(予約)			
3	(予約)			
4	MAC Address	Get	UINT8の配列	デバイスのMACアドレス
5	Interface Counters	Get	UINT32 の配列	MIB-IIインターフェースカウンターを含む配列 (rfc1213) 配列インデックスについては下表を参照してください。
6	(予約)			

インスタンスアトリビュート (インスタンス#2～#3)

#	名前	アクセス	データ型	説明
1～4	(予約)			
5	Interface Counters	Get	UINT32 の配列	MIB-IIインターフェースカウンターを含む配列 (rfc1213) 配列インデックスについては下表を参照してください。
6	Media Counters	Get	UINT32 の配列	ポート用のEthernet Like MIBカウンターを含む配列。配列インデックスについては下表を参照してください。

Interface Counters

: Interface Countersアトリビュート (#5) の配列インデックス

インデックス	Name	説明
0	In Octets	インターフェースで受信されたオクテット数
1	In Unicast Packets	インターフェースで受信されたユニキャストパケット数
2	In Non-Unicast Packets	インターフェースで受信された非ユニキャスト (マルチキャスト/ブロードキャスト) パケット数
3	In Discards	インターフェースで受信されたが破棄されたインバウンドパケット数
4	In Errors	エラーを含むインバウンドパケット数 (In Discardsに含まれない)
5	In Unknown Protos	不明なプロトコルのインバウンドパケット数
6	Out Octets	インターフェースで送信されたオクテット数
7	Out Unicast packets	インターフェースで送信されたユニキャストパケット数
8	Out Non-Unicast Packets	インターフェースで送信された非ユニキャスト (マルチキャスト/ブロードキャスト) パケット数
9	Out Discards	破棄されたアウトバウンドパケット数
10	Out Errors	エラーを含むアウトバウンドパケット数

Media Counters

: Media Countersアトリビュート (#6) の配列インデックス

インデックス	Name	説明
0	AlignmentErrors;	長さが8進数整数以外の受信フレーム数
1	FCSErrors;	FCSチェックにエラーした受信フレーム数
2	SingleCollisions;	一回だけコリジョンが発生して伝送に成功したフレーム数
3	MultipleCollisions;	二回以上コリジョンが発生して伝送に成功したフレーム数
4	SQETestErrors;	SQEテストエラーが生成された回数
5	DeferredTransmissions;	第一回の伝送試行が媒体ビジーにより遅延しているフレーム数
6	LateCollisions;	パケット伝送に対して512ビット目より後でコリジョンが検出された回数
7	ExcessiveCollisions;	コリジョンが多すぎて伝送に失敗したフレーム数
8	IMACTransmitErrors;	内部MACのサブレイヤー送信エラーにより伝送が失敗したフレーム数
9	ICarrieSenseErrors;	フレームの送信試行時に搬送波検出条件が損失した、またはアサートされなかった回数
10	IFrameTooLong;	最大許容フレームサイズを超過した受信フレーム数
11	IMACRecieveErrors;	MACの内部サブレイヤー受信エラーにより、インターフェース上での受信に失敗したフレーム数

11.10 CIPポートコンフィグレーションオブジェクト (0Dh)

カテゴリ

拡張

オブジェクトの説明

このオブジェクトは、ネットワーク側にCIPポートオブジェクト ([ポートオブジェクト\(F4h\)](#), [ページ 104](#)を参照) を設定して列挙するために使用されます。基本的に、これはホストアプリケーションの中でCIP ポートを表すインスタンスとアトリビュートを作成および更新することです。このプロセスは、非コネクション型CIPルーティングのサポートが有効にされたときに必要になります ([EtherNet/IPホストオブジェクト \(F8h\)](#), [ページ 175](#)、インスタンスアトリビュート#17を参照)。

このオブジェクトの中の各インスタンスはCIP ポートオブジェクトの中のインスタンスに対応します。このオブジェクトは最大8個のインスタンスをサポートし、ここではインスタンス#1 がローカルTCP ポート専用であり、ホストアプリケーションは追加で7個までのポートを実装することを可能にします。インスタンス#1には自動的にデフォルト値が代入されますが、ホストアプリケーションはインスタンスアトリビュート#2と#4をカスタム化することもできます。

アトリビュート#7を除き、セットアップの間だけインスタンスアトリビュートへの書き込みが可能です。ホストアプリケーションはホストアプリケーションの中にあるすべてのポートに関してインスタンスアトリビュート #7が更新され続けるようにします。



モジュールがホストアプリケーションが受け持つエラー制御機能を引き受けず、モジュールがホストアプリケーションによって設定されたデータの正しさを検証もしないことに注意してください。

サポートされているコマンド

オブジェクト : Get_Attribute
 Create
 Delete

インスタンス : Get_Attribute
 Set_Attribute

オブジェクトアトリビュート (インスタンス#0)

#	名前	アクセス	データ型	値
1	Name	Get	CHARの配列	「CIP Port Configuration」
2	Revision	Get	UINT8	01h
3	Number of instances	Get	UINT16	-
4	Highest instance no.	Get	UINT16	-
11	Max. no. of instances	Get	UINT16	0008h

インスタンスアトリビュート (インスタンス#1)

#	名前	アクセス	データ型	説明
1	Port Type	Set	UINT16	ポートを列挙します (CIP仕様を参照。 www.odva.org から入手可能)
2	Port Number	Set	UINT16	このポートと関連付けられたCIPポート番号
3	Link Path	Set	UINT8の配列	このポートに対するオブジェクトを識別する論理パスセグメント。
4	Port Name	Set	CHARの配列	ポートを命名する文字列 (最大文字数は64)。
5	-	-	-	(予約)
6	-	-	-	(予約)
7	Node Address	Set	UINT8の配列	ポートにおけるこのデバイスのノード番号。データ型は範囲をポートセグメントに制限します。エンコードされたポート番号はアトリビュート #2に指定された値に一致しなければなりません。 ポートにノード番号を持たないデバイスは、ポートセグメントの中の長さゼロのノードアドレス(すなわち10h 00h)に指定できます。 ノードアドレスがランタイム中に変化した場合、ホストアプリケーションはこのアトリビュートの更新も受け持ちます。
8	Port Node Range	Set	構造体: UINT16 (最小) UINT16 (最大)	ポートでの最小と最大ノード番号。 このアトリビュートのサポートは条件次第です。ノード番号がデータ型の範囲内 (例えば、DeviceNet) で報告され得るという条件で、アトリビュートはサポートされなければなりません。そうでない場合 (4バイトのIPアドレスを使用する EtherNet/IPネットワークなど)、アトリビュートはサポートされません。

11.11 ファンクショナルセーフティモジュール・オブジェクト (11h)

カテゴリ

拡張

オブジェクトの説明

このオブジェクトはAnybus CompactCom モジュールと接続されたセーフティモジュールによって提供される情報を格納します。下記のアトリビュート値に関しては使用されるセーフティモジュールの説明書を参照ください。

サポートコマンド

オブジェクト : Get_Attribute
 Error_Confirmation
 Set_IO_Config_String
 Get_Safety_Output_PDU
 Get_Safety_Input_PDU

インスタンス : Get_Attribute

オブジェクトアトリビュート (インスタンス#0)

#	名前	アクセス	データ型	値
1	Name	Get	CHARの配列	「Functional Safety Module」
2	Revision	Get	UINT8	01h
3	Number of instances	Get	UINT16	0001h
4	Highest instance no.	Get	UINT16	0001h

インスタンスアトリビュート (インスタンス#1)

#	名前	アクセス	データ型	説明
1	State	Get	UINT8	セーフティモジュールの現在の状態 使用されるセーフティモジュールの説明書を参照ください。
2	Vendor ID	Get	UINT16	セーフティモジュールのベンダーID。 例0001h (HMS Industrial Networks) 使用されるセーフティモジュールの説明書を参照ください。
3	IO Channel ID	Get	UINT16	セーフティモジュールが取り付けられているIOチャンネルを表す。 使用されるセーフティモジュールの説明書を参照ください。
4	Firmware version	Get	構造体 UINT8 (メジャー) UINT8 (マイナー) UINT8 (ビルド)	セーフティモジュールのファームウェアバージョン。 フォーマット: バージョン「2.18.3」は、1バイト目 = 0x02、2バイト目 = 0x12、3バイト目 = 0x03として表されます。
5	Serial number	Get	UINT32	製造時にセーフティモジュールに割り当てられた、32ビットの番号。 使用されるセーフティモジュールの説明書を参照ください。
6	Output data	Get	UINT8の配列	セーフティモジュールの出力データ (ネットワークからのデータ) の現在値 注: このデータはAnybus CompactCom モジュールから提供されるので安全ではありません。
7	Input data	Get	UINT8の配列	セーフティモジュールの入力データ (ネットワークへのデータ) の現在値 注: このデータはAnybus CompactCom モジュールから提供されるので安全ではありません。
8	Error counters	Get	構造体 UINT16 (ABCC DR) UINT16 (ABCC SE) UINT16 (SM DR) UINT16 (SM SE)	エラー カウンター (各カウンターはFFFFhで計数を停止) ABCC DR: Anybus CompactCom モジュールによって破棄されたセーフティモジュールからの (予期せぬ) 応答。 ABCC SE: Anybus CompactCom モジュールにより検出されたシリアル受信エラー。 SM DR: セーフティモジュールによって棄却された、Anybus CompactComモジュールからの (期待していない) 応答。 SM SE: セーフティモジュールによって検出されたシリアル受信エラー。
9	Event log	Get	UINT8の配列	最後に発生したセーフティモジュールのイベント情報 (存在する場合) が、このアトリビュートに記録されます。それより以前のイベント情報は新たなイベントが記録されるときに消去されます。 HMSサポートによる評価で使用されます。
10	Exception information	Get	UINT8	Anybusオブジェクトの例外コードが「セーフティ通信エラー」(09h)に設定されている場合、追加の例外情報がここに表されます (下表を参照)。
11	Bootloader version	Get	構造体 UINT8 メジャー UINT8 マイナー	セーフティモジュールのブートローダーバージョン。 フォーマット: バージョン「2.12」は、1バイト目 = 0x02、2バイト目 = 0x0Cとして表されます。
12	Vendor block safe uc1	Get	UINT8の配列	セーフティモジュールは、追加のベンダー固有データをAnybus CompactComに供給する場合があります。そのようなデータが使用可能な場合、このアトリビュートに表されます。
13	Vendor block safe uc2	Get	UINT8の配列	セーフティモジュールは、追加のベンダー固有データをAnybus CompactComに供給する場合があります。そのようなデータが使用可能な場合、このアトリビュートに表されます。

例外情報

例外コード09hがAnybusオブジェクトにセットされている場合、アプリケーション内のファンクショナルセーフティモジュールに関するエラーが発生しています。例外情報はこの表に従ってインスタンスアトリビュート #10に表されます：

値	Exception Information
00h	情報なし
01h	ポーレートがサポートされていない
02h	開始メッセージが存在しない
03h	予期せぬメッセージ長
04h	応答に予期せぬコマンドが含まれている
05h	予期せぬエラーコード
06h	セーフティアプリケーションが見つからない
07h	セーフティアプリケーションのCRCが無効
08h	フラッシュアクセスなし
09h	ブートローダー通信中に間違ったセーフティプロセッサから応答があった
0Ah	ブートローダーのタイムアウト
0Bh	ネットワーク固有のパラメータエラー
0Ch	無効なIOコンフィグレーション文字列
0Dh	セーフティマイクロプロセッサ間で応答が異なる (モジュールタイプが異なるなど)
0Eh	互換性のないモジュール (サポートしているネットワークなど)
0Fh	(CRCエラーなどにより) 最大回数の再送が行われた
10h	ファームウェア ファイル エラー
11h	ファンクショナルセーフティ・ホストオブジェクトのアトリビュート #4にある周期時間の値は現在のポーレートでは使用できない
12h	起動テレグラムのSPDU入力サイズが無効
13h	起動テレグラムのSPDU出力サイズが無効
14h	不正形式の入力SPDU
15h	Anybusからセーフティモジュールへの初期化エラー

コマンドの詳細: Error_Confirmation

カテゴリ

拡張

詳細

コマンドコード	10h
有効な対象:	オブジェクト

説明

何らかの理由によりセーフティモジュールがセーフ状態になった場合、そのモジュールがセーフ状態から抜けるにはエラー確認を受信する必要があります。このコマンドを使用すると、何らかの理由で同時にセーフ状態になっているセーフティモジュールのすべてのセーフティチャネルをリセットすることが可能です。オペレータなどによってエラーがクリアされると、アプリケーションがこのコマンドをAnybus CompactComモジュールに発行します。Anybus CompactComはこのコマンドをセーフティモジュールに転送します。

チャネルのセーフ状態は、安全PLCまたはセーフティモジュールによっても確認することができます。

このコマンドで確認できます。

- コマンド詳細
(データなし)
- 応答詳細
(データなし)

コマンドの詳細: Set_IO_Config_String

カテゴリ

拡張

詳細

コマンドコード	11h
有効な対象:	オブジェクト

説明

このコマンドは、セーフティ入出力のデフォルト コンフィグレーションを変更する必要があるとき、ホストアプリケーションから送られます。この文字列は、コンフィグレーションをセーフティモジュールに提供するためにその他の手段(例: PLCやその他のツール)が存在しないネットワークによって使用されます。詳細はセーフティモジュールの仕様書を参照ください。渡された文字列はHMSにより生成されており、このコマンドを使用してそのまま渡される必要があります。

この文字列についての情報は、文字列の送り先となるセーフティモジュールの仕様書を参照ください。

- コマンド詳細

フィールド	内容
CmdExt[0]	(未使用)
CmdExt[1]	
Data[0 ~ n]	データ(バイト文字列) このデータはIO コンフィグレーション文字列から成っており、そのデータフォーマットはセーフティ ネットワークに依存します。

- 応答詳細

(データなし)

コマンドの詳細: Get_Safety_Output_PDU

カテゴリ

拡張

詳細

コマンドコード	12h
有効な対象:	オブジェクト

説明

このコマンドは、PLCによって送信された完全な安全PDU出力を取得するために、アプリケーションから発行できます。Anybus CompactCom 40 EtherNet/IPは完全な安全PDUで応答します。アプリケーションはこれを解釈する必要があります。

- コマンド詳細
(データなし)
- 応答詳細

フィールド	内容
CmdExt[0]	(未使用)
CmdExt[1]	
Data[0 ~ n]	PLCからの安全PDU

コマンドの詳細: Get_Safety_Input_PDU

カテゴリ

拡張

詳細

コマンドコード	13h
有効な対象:	オブジェクト

説明

このコマンドは、セーフティモジュールによって送信された完全な安全PDU入力を取得するために、アプリケーションから発行できます。Anybus CompactCom 40 EtherNet/IPは完全な安全PDUで応答します。アプリケーションはこれを解釈する必要があります。

- コマンド詳細
(データなし)
- 応答詳細

フィールド	内容
CmdExt[0]	(未使用)
CmdExt[1]	
Data[0 ~ n]	セーフティモジュールからの安全PDU

オブジェクト固有のエラーコード

エラーコード	説明	コメント
01h	セーフティモジュールはメッセージを拒否しました。	セーフティモジュールによって送られたエラーコードはMsgData[2]とMsgData[3]の中にあります。
02h	セーフティモジュールからのメッセージ応答のフォーマットが不正です (例、不正な長さ)。	-

12 ホストアプリケーションオブジェクト

12.1 基本情報

この章では、本モジュールで実装されているホストアプリケーションオブジェクトについて説明します。以下に示すオブジェクトをホストアプリケーションのファームウェアにオプションで実装することで、EtherNet/IPの機能を拡張することができます。

標準オブジェクト:

- [Energy Control オブジェクト \(F0h\), ページ 169](#)
- [アセンブリマッピングオブジェクト \(EBh\) - \(『Anybus CompactCom 40 Software Design Guide』を参照\)](#)
- [モジュラーデバイスオブジェクト \(ECh\) - \(『Anybus CompactCom 40 Software Design Guide』を参照\)](#)
- [SYNCオブジェクト \(EEh\), ページ 168](#)
- [Energy Reporting オブジェクト \(E7h\), ページ 162](#)
- [アプリケーションデータオブジェクト \(FEh\) - \(『Anybus CompactCom 40 Software Design Guide』を参照\)](#)
- [アプリケーションオブジェクト \(FFh\) - \(『Anybus CompactCom 40 Software Design Guide』を参照\)](#)

ネットワーク固有のオブジェクト:

- [ファンクショナルセーフティオブジェクト \(E8h\), ページ 163](#)
- [アプリケーション・ファイルシステムインターフェース・オブジェクト \(EAh\), ページ 165](#)
- [CIP Identity ホストオブジェクト \(EDh\), ページ 166](#)
- [EtherNet/IPホストオブジェクト \(F8h\), ページ 175](#)
- [Ethernet ホストオブジェクト \(F9h\), ページ 184](#)

12.2 Energy Reporting オブジェクト (E7h)

カテゴリ

拡張

オブジェクトの説明

ホストアプリケーションはこのオブジェクトを使用して、消費または生成された電力を報告する方法を標準化します。このオブジェクトのレポート機能は限定的なものです。より詳細なレポート機能を提供するネットワークでは、透過性のある方法でホストアプリケーション側に実装する必要があります。

サポートコマンド

オブジェクト : Get_Attribute

インスタンス : Get_Attribute

オブジェクトアトリビュート (インスタンス#0)

#	名前	アクセス	データ型	値
1	Name	Get	CHARの配列	「Energy Reporting」
2	Revision	Get	UINT8	01h
3	Number of instances	Get	UINT16	0001h
4	Highest instance no.	Get	UINT16	0001h

インスタンスアトリビュート (インスタンス#1)

拡張

#	名前	アクセス	種類	説明
1	Energy Reading	Get	構造体: UINT32 UINT32	アプリケーションによって消費または生成されるエネルギー量 (Wh)。不揮発性メモリに格納。 最初のUINT32はEnergy Readingの下位部分を表し、2番目のUINT32はEnergy Readingの上位部分を表す。
2	Direction	Get	BOOL	ホストが電力を消費しているか生成しているかを示す。 値: 意味: 0: 生成 1: 消費
3	Accuracy	Get	UINT16	精度範囲は0.01%の読み取り値 0: 不明
4	Current Power Consumption	Get	UINT16	現在の消費電力 (単位 : 定格消費電力の0.01%)
5	Nominal Current Consumption	Get	UINT32	定格消費電力 (単位 : mW)

12.3 ファンクショナルセーフティオブジェクト (E8h)

カテゴリ

拡張

オブジェクトの説明



このオブジェクトはセーフティモジュールが未使用のときは実装しないでください。

このオブジェクトはアプリケーションのセーフティ設定を指定します。ファンクショナルセーフティのサポートが必要であり、セーフティモジュールをAnybus CompactCom モジュールと接続する場合は必須です。

サポートコマンド

オブジェクト : Get_Attribute

インスタンス : Get_Attribute

オブジェクトアトリビュート (インスタンス#0)

#	名前	アクセス	データ型	値
1	Name	Get	CHARの配列	「Functional Safety」
2	Revision	Get	UINT8	01h
3	Number of instances	Get	UINT16	0001h
4	Highest instance no.	Get	UINT16	0001h

インスタンスアトリビュート (インスタンス#1)

#	名前	アクセス	データ型	デフォルト値	コメント
1	Safety enabled	Get	BOOL	-	TRUEのときセーフティモジュールと通信が可能です。 注: ファンクショナルセーフティがサポートされない場合、このアトリビュートをFALSEに設定しなければなりません。
2	Baud Rate	Get	UINT32	1020 kbit/s	このアトリビュートはAnybus CompactComとセーフティモジュール間の通信ボーレート (bits/s) を設定します。 有効値: <ul style="list-style-type: none"> • 625 kbit/s • 1000 kbit/s • 1020 kbit/s (デフォルト) このアトリビュートに他の値を設定すると、モジュールがEXCEPTIONステートになる原因になります。 このアトリビュートはオプションです。実装されない場合、デフォルト値が使用されます。 注: IXXAT Safe T100を使用するときは、ホストアプリケーションはこのアトリビュートを実装してはいけません。
3	(予約)				

#	名前	アクセス	データ型	デフォルト値	コメント
4	Cycle Time	Get	UINT8	-	<p>Anybusとセーフティモジュール間の通信周期時間 (ミリ秒)。 注: IXXAT Safe T100を使用するときは、ホストアプリケーションはこのアトリビュートを実装してはいけません。 有効値:</p> <ul style="list-style-type: none"> • 2 ms • 4 ms • 8 ms • 16 ms <p>他の値がこのアトリビュートに設定されると、AnybusはExceptionステートに入ります。 このアトリビュートはオプションです。実装されない場合、選択されたボーレートに対応した最小周期時間が適用されます:</p> <ul style="list-style-type: none"> • 1020 kbit/sの場合は 2 ms • 1000 kbit/sの場合は 2 ms • 625 kbit/sの場合は 4 ms <p>Anybus CompactComは上記の最小値に従って周期時間を検証します。例えばボーレートが625 kbit/sで周期時間が2 msに設定されていると、Anybus CompactComはEXCEPTION (例外) 状態になります。</p>
5	FW upgrade in progress	Set	BOOL	False	<p>接続されているセーフティモジュールのファームウェアをAnybus CompactComがアップグレードしているかどうかを示します。これは、Anybus CompactComが通常よりも長くNW_INIT状態にとどまることを意味します。</p>

12.4 アプリケーション・ファイルシステムインターフェース・オブジェクト (EAh)

カテゴリ

拡張

オブジェクトの説明

このオブジェクトは内蔵ファイルシステムとのインターフェースになります。各インスタンスはファイルストリームへのハンドルを表し、ファイルシステム操作のためのサービスを格納しています。これでユーザーはFTPサーバを介してアプリケーションソフトウェアをダウンロードできます。アプリケーションが利用可能なメモリ空間を判断します。

このオブジェクトについての完全な詳細は、*Anybus CompactCom 40 Software Design Guide*に記載されています。

12.5 CIP Identity ホストオブジェクト (EDh)

カテゴリ

拡張

オブジェクトの説明

このオブジェクトによってアプリケーションは追加のCIP Identity インスタンスに対応できるようになります。追加の製品識別情報について情報を提供するために使用されます (例えばインストールされているソフトウェアについて情報)。

CIP Identity オブジェクトにおいて最初のインスタンスは動作が変わりません。CIP Identity ホストオブジェクトにインスタントを実装するとき、インスタンス2から開始されるCIP Identity オブジェクトにマッピングされます。CIP Identity ホストオブジェクトにあるインスタンスno. 1は、CIP identity オブジェクトのインスタンスno. 2等にマッピングされます。

下記も参照してください。

- [Identity オブジェクト \(01h\)](#), ページ 78 (CIPオブジェクト)

サポートされているコマンド

オブジェクト : Get_Attribute
 インスタンス : Get_Attribute
 Get_Attribute_All

オブジェクトアトリビュート (インスタンス#0)

#	名前	アクセス	データ型	値
1	Name	Get	CHARの配列	「CIP Identity」
2	Revision	Get	UINT8	01h
3	Number of instances	Get	UINT16	アプリケーション依存
4	Highest instance no.	Get	UINT16	アプリケーション依存

インスタンスアトリビュート (インスタンス#1)

#	名前	アクセス	データ型	説明
1	Vendor ID	Get	UINT16	これらの値はCIPアイデンティティオブジェクトのインスタンス#2以降の値を置き換えます。 下記も参照してください。 Identity オブジェクト (01h) , ページ 78 (CIPオブジェクト)
2	Device Type	Get	UINT16	
3	Product Code	Get	UINT16	
4	Revision	Get	構造体: UINT8 Major UINT8 Minor	
5	Status	Get	UNIT16	
6	Serial Number	Get	UINT32	
7	Product Name	Get	CHARの配列	

コマンドの詳細: Get_Attribute_All

カテゴリ

拡張

詳細

コマンドコード: 10h

有効な対象: オブジェクト

説明

このサービスはCIP Identity ホストオブジェクトに存在するすべてのインスタンスについてアプリケーションによって実装される必要があります。Identity (識別) データがネットワークから要求されると、Anybusモジュールはこのコマンドをアプリケーションに発行します。アプリケーションはこれに、要求されたインスタンスの全アトリビュートから成る構造体を含むメッセージで応答します。

- コマンド詳細
(データなし)
- 応答詳細

フィールド	内容	コメント
MsgData[0, 1]	Vendor ID	ABCC CIPアイデンティティデータ
MsgData[2, 3]	Device type	
MsgData[4, 5]	Product code	
MsgData[6]	Major revision	
MsgData[7]	Minor revision	
MsgData[8,9]	状態	
MsgData[10 .. .13]	Serial number	
MsgData[14 n]	Product name	

12.6 SYNCオブジェクト (EEh)

カテゴリ

拡張

オブジェクトの説明

このオブジェクトは、データを消費するIOコネクションが最後に確立された際の周期時間を格納するためにのみ使用されます。

サポートされているコマンド

オブジェクト : Get_Attribute
 インスタンス : Get_Attribute
 Set_Attribute

オブジェクトアトリビュート (インスタンス#0)

(詳細については、一般的な *Anybus CompactCom 40 Software Design Guide* を参照してください。)

インスタンスアトリビュート (インスタンス#1)

アトリビュートは EtherNet/IP においては次のように表されます。

#	名前	アクセス	データ型	説明
1	Cycle time	Get/Set	UINT32	データを消費するIOコネクションが最後に確立された際のRPI (O→T RPI)
2-8	(未実装)			

12.7 Energy Control オブジェクト (F0h)

カテゴリ

拡張

オブジェクトの説明

このオブジェクトを使用すると、ホストアプリケーションにEnergyコントロール機能 (電力固有の設定) が実装されます。このオブジェクトの実装は任意です。すべてのインスタンスアトリビュートが必要であるとされているため、アプリケーションに実装する必要があります。実行時にアトリビュートが欠落していることをAnybusモジュールが検出した場合、適切なネットワークエラーが送信されて、Anybusオブジェクトのインスタンスアトリビュートであるエラーカウンターで「破棄された応答」のカウンターが増加します。

オブジェクトで利用可能な各インスタンスは、省電力モードに対応しています。利用可能なモードの数は装置によって異なります。それらのモードはアプリケーションにて定義する必要があります。インスタンスの番号が大きいほど、より多くの電力を節約できます。最も番号が大きいインスタンスは「電源オフ」に相当します。すなわち、装置は基本的にシャットダウンされます。オブジェクトのインスタンス1は「動作可能」を表します。すなわち、装置は完全に機能し、電力は全く節約されません。そのため、意味のある実装では、常に2つのインスタンス、すなわち、省電力用のインスタンスと動作用のインスタンスを持ちます。このオブジェクトがPROFINET用に実装されている場合は、少なくとも次の3つのインスタンスが必要です。「動作可能」、「省電力モード1」、「電源オフ」。

インスタンスの最大数は8です。なお、これらのモードは常に存在しており、動的に作成されたり削除されたりすることはありません。インスタンスのリストに穴を空けておく (サポートされていないインスタンスが途中で存在する) ことは許容されません。

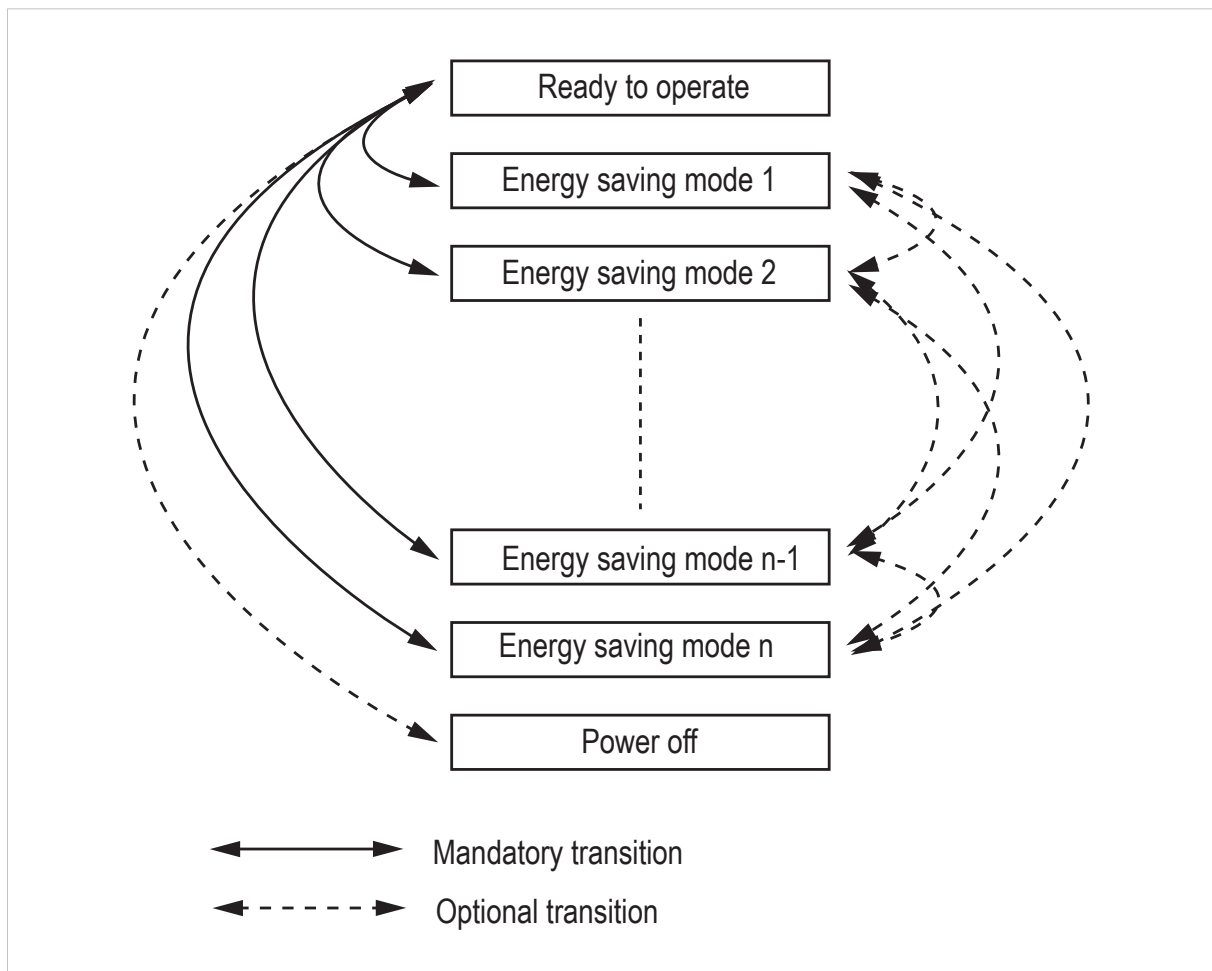


図 7

サポートコマンド

オブジェクト : Get_Attribute
 StartPause
 EndPause
 Preview_Pause_Time (PROFINETではありません)

インスタンス : Get_Attribute

オブジェクトアトリビュート (インスタンス#0)

#	名前	アクセス	データ型	値
1	Name	Get	CHARの配列	「Energy Control」
2	Revision	Get	UINT8	02h
3	Number of instances	Get	UINT16	-
4	Highest instance no.	Get	UINT16	作成されたインスタンス番号のうち最も大きい番号。最大値は8です。
11	Current Energy Saving Mode	Get	UINT16	現在使用されている省電力モードのインスタンス番号。モード移行中は新しい省電力モードが表示されます。「動作可能」はインスタンス#1に相当し、「電源オフ」は最大インスタンス番号に相当します。
12	Remaining time to destination	Get	UINT32	モードを変更したとき、このパラメータは移行が完了するまでの実際の残り時間 (単位：ミリ秒) を反映します。 値を動的に生成できない場合、モードの遷移元から遷移先への移行に関する静的な値が使用されます。 この値が無限大または不明な場合、最大値0xFFFFFFFFが使用されます。
13	Energy consumption to destination	Get	FLOAT	モードを変更したとき、このパラメータは移行が完了するまでに実際に消費される電力 (単位：kWh) を反映します。 値を動的に生成できない場合、モードの遷移元から遷移先への移行に関する静的な値が使用されます。 値が定義されていない場合、0.0という値が使用されます。
14	Transition to 「Power off」 mode supported	Get	BOOL	「電源オフ」モードへの移行がサポートされているか否かを示します。 0: サポートなし 1: サポートあり

インスタンスアトリビュート (インスタンス#1 ~ #8)

#	名前	アクセス	データ型	説明
1	ModeAttributes	Get	BITS16	<p>ビット0: 意味:</p> <p>0: 静的時間および電力値のみ使用可能 (アトリビュートのビット0の値は実装されていない)</p> <p>1: 時間と電力に関する動的な値を使用可能</p> <p>Bit 1 ~ 15: 予約</p>
2	TimeMinPause	Get	UINT32	ミリ秒で表される最小休止時間。 (t_{pause}) この値が無限大または不明な場合、最大値0xFFFFFFFFが使用されます。
3	TimeToPause	Get	UINT32	この省電力モードに移行するまでの最大所要時間。(ミリ秒、 t_{off}) この値が無限大または不明な場合、最大値0xFFFFFFFFが使用されます。
4	TimeToOperate	Get	UINT32	「動作可能」モードに移行するための最大所要時間。(ミリ秒、 t_{on}) この値が無限大または不明な場合、最大値0xFFFFFFFFが使用されます。
5	TimeMinLengthOfStay	Get	UINT32	装置がこのモードを維持しなければならない最小時間。単位はミリ秒です。 (ミリ秒、 $t_{\text{off_min}}$) この値が無限大または不明な場合、最大値0xFFFFFFFFが使用されます。
6	TimeMaxLengthOfStay	Get	UINT32	このモードを維持できる最大時間。単位はミリ秒です。 最大値が使用できない場合、または実装されていない場合は、最大値0xFFFFFFFFが使用されます。
7	ModePowerConsumption	Get	FLOAT	このモードで消費される電力量。(kW) 値が定義されていない場合、0.0という値が使用されます。
8	EnergyConsumptionToPause	Get	FLOAT	このモードへ移行するのに必要な電力量。(kWh) 値が定義されていない場合、0.0という値が使用されます。
9	EnergyConsumptionToOperate	Get	FLOAT	このモードから「動作可能」モードに移行するのに必要な電力量。(kWh) 値が定義されていない場合、0.0という値が使用されます。
10	Availability	Get	BOOL	現在のデバイスの状態において、この省電力モードが使用可能か否かを示します。 PROFINETには使用されません。 False 使用不可 True 使用可能 (アトリビュートが実装されていない場合は値)
11	Power Consumption	Get	UINT32	この状態にあるときのデバイスの消費電力を示します。 PROFINETには使用されません。

コマンドの詳細: Start_Pause

詳細

コマンドコード 10h
 有効な対象 : オブジェクト

説明

このコマンドは、システムがシステムの休止を開始したいときに、ホストアプリケーションに送られます。休止時間はミリ秒単位で指定します。メッセージの応答には、移行先のモード (選択した省電力モードのインスタンス番号) が含まれません。

• コマンド詳細

フィールド	内容	コメント
Data[0]	休止時間 (下位ワード、下位バイト)	休止時間 (ms)
Data[1]	休止時間 (下位ワード、上位バイト)	
Data[2]	休止時間 (上位ワード、下位バイト)	
Data[3]	休止時間 (上位ワード、上位バイト)	

• 応答詳細

フィールド	内容	コメント
Data[0]	インスタンス番号 (下位バイト)	選択した電力モードのインスタンス番号
Data[1]	インスタンス番号 (下位バイト)	

要求された休止時間の間にアプリケーションが状態を選択できない場合、下表のエラーコードのいずれかを返しません。

#	エラーコード	説明
0x0D	Invalid state	現在のデバイスの状態と要求された休止時間では、何らかの省電力モードに入ることができません
0x12	Value too low	要求された休止時間が短すぎます

コマンドの詳細: End_Pause

詳細

コマンドコード 11h
 有効な対象 : オブジェクト

説明

システムが休止モードから「動作可能」モードに戻したいとき、このコマンドがホストアプリケーションに送信されます。応答メッセージで、スイッチを有効化するために必要なミリ秒数が返されます。

- コマンド詳細
 (なし)
- 応答詳細

フィールド	内容	コメント
Data[0]	動作するまでの時間 (下位ワード、下位バイト)	「動作可能」に切り替わるまでの時間
Data[1]	動作するまでの時間 (下位ワード、上位バイト)	
Data[2]	動作するまでの時間 (上位ワード、下位バイト)	
Data[3]	動作するまでの時間 (上位ワード、上位バイト)	

アプリケーションが休止状態を終了できない場合、下表のエラーコードを返します。

#	エラーコード	説明
0x0D	Invalid state	現在のデバイスの状態では、休止の終了は今のところ不可能です

コマンドの詳細: Preview_Pause_Time

詳細

コマンドコード 12h
 有効な対象 : オブジェクト

説明

PROFINETデバイスでは使用されません。

このコマンドは、アプリケーションの省電力モードの選択をシステムがプレビューしたいときに、ホストアプリケーションに送られます。休止時間はミリ秒単位で指定します。応答には、StartPauseサービスが送られていればアプリケーションが選択していたであろう移行先モード (すなわち選択した省電力モードのインスタンス番号) が含まれます。省電力モードへの移行は行われません。

• コマンド詳細

フィールド	内容	コメント
Data[0]	休止時間 (下位ワード、下位バイト)	休止時間 (ms)
Data[1]	休止時間 (下位ワード、上位バイト)	
Data[2]	休止時間 (上位ワード、下位バイト)	
Data[3]	休止時間 (上位ワード、上位バイト)	

• 応答詳細

フィールド	内容	コメント
Data[0]	インスタンス番号 (下位バイト)	選択した電力モードのインスタンス番号
Data[1]	インスタンス番号 (下位バイト)	

要求された休止時間の間にアプリケーションが状態を選択できない場合、下表のエラーコードのいずれかを返しません。

#	エラーコード	説明
0x0D	Invalid state	現在のデバイスの状態と要求された休止時間では、何らかの省電力モードに入ることができません
0x12	Value too low	要求された休止時間が短すぎます

12.8 EtherNet/IPホストオブジェクト (F8h)

カテゴリ

基本、拡張

オブジェクトの説明

このオブジェクトを使用することにより、ホストアプリケーションにEtherNet/IP固有の機能が実装されます。このオブジェクトは、イーサネットホストオブジェクトとは異なるものですので混同しないでください。詳しくは[Ethernet ホストオブジェクト \(F9h\)](#), ページ 184を参照してください。

このオブジェクトは任意に実装できます。これにより、ホストアプリケーションが以下のアトリビュートを全くサポートしないか、一部のアトリビュートをサポートするか、すべてのアトリビュートをサポートするかを選択できます。モジュールは起動時にこれらのアトリビュートの値を取得しようと試みます。値を取得しようとしたアトリビュートがホストアプリケーションに実装されていない場合、エラーメッセージ (06h、「Invalid CmdExt[0]」) を返します。そのような場合、モジュールはデフォルト値を使用します。

モジュールが以下に示されていないアトリビュートの値を取得しようとした場合、エラーメッセージ (06h、「Invalid CmdExt[0]」) を返します。

このオブジェクトにアクセスするために使用されるコマンドの一部には、分割が必要な場合があります。詳細については、[メッセージの分割](#), ページ 142を参照してください。

モジュールがEIP QuickConnect機能を使用するように設定されている場合、EDSファイルの変更が必要です。EDSファイルが変更されると共に、モジュールの識別情報の変更、そしてモジュールの認証も必要になります。

下記も参照してください。

- [Identity オブジェクト \(01h\)](#), ページ 78 (CIPオブジェクト)
- [アセンブリオブジェクト \(04h\)](#), ページ 82 (CIPオブジェクト)
- [ポートオブジェクト \(F4h\)](#), ページ 104 (CIPオブジェクト)
- [CIPポートコンフィグレーションオブジェクト \(0Dh\)](#), ページ 152
- 『Anybus CompactCom 40 Software Design Guide』、「エラーコード」

サポートされているコマンド

オブジェクト : Get_Attribute
 Process_CIP_Object_Request
 Set_Configuration_Data
 Process_CIP_Routing_Request
 Get_Configuration_Data

インスタンス : Get_Attribute

オブジェクトアトリビュート (インスタンス#0)

#	名前	アクセス	データ型	値
1	Name	Get	CHARの配列	「EtherNet/IP」
2	Revision	Get	UINT8	02h
3	Number of instances	Get	UINT16	0001h
4	Highest instance no.	Get	UINT16	0001h

インスタンスアトリビュート (インスタンス#1)

基本

#	名前	アクセス	データ型	デフォルト値	コメント
1	Vendor ID	Get	UINT16	005Ah	これらの値は起動時に Identity オブジェクト (CIP) に設定されます。 下記も参照してください。 <ul style="list-style-type: none"> • ネットワークID, ページ 11 • Identity オブジェクト (01h), ページ 78 これらいずれかのアトリビュート変更の際には、新たなベンダーIDが必要になることにご注意ください。
2	Device Type	Get	UINT16	002Bh	
3	Product Code	Get	UINT16	0037h	
4	Revision	Get	構造体: UINT8 Major UINT8 Minor	(ソフトウェアリビジョン)	
5	Serial Number	Get	UINT32	(製造時の設定)	
6	Product Name	Get	CHARの配列	「Anybus CompactCom 40 EtherNet/IP(TM)」	

拡張

#	名前	アクセス	データ型	デフォルト値	コメント
7	Producing Instance No.	Get	UINT16の配列	-	この配列中の値は、ホストアプリケーションのアセンブリマッピングオブジェクトインスタンスのアトリビュート#11 (Write PD Instance List) にリストされているものと一致する、EtherNet/IPアセンブリインスタンス番号です。アセンブリマッピングオブジェクトが実装されていない場合、producingインスタンス番号を設定するためにこの配列に許可される要素は一つです。 配列におけるエントリの最大数は6です。 例については、後述の「複数アセンブリインスタンス」を参照してください。
8	Consuming Instance No.	Get	UINT16の配列	-	この配列中の値は、ホストアプリケーションのアセンブリマッピングオブジェクトインスタンスのアトリビュート#12 (Read PD Instance List) にリストされているものと一致する、EtherNet/IPアセンブリインスタンス番号です。アセンブリマッピングオブジェクトが実装されていない場合、consumingインスタンス番号を設定するためにこの配列に許可される要素は一つです。 配列におけるエントリの最大数は6です。 例については、後述の「複数アセンブリインスタンス」を参照してください。
9	Enable communication settings from Net	Get	BOOL	True	値 意味 True ネットワークから設定可能 False ネットワークから設定不可 下記も参照してください。 <ul style="list-style-type: none"> • TCP/IPインターフェースオブジェクト (F5h), ページ 106 (CIPオブジェクト) • Ethernet リンクオブジェクト (F6h), ページ 109 (CIPオブジェクト) • ネットワークコンフィグレーションオブジェクト (04h), ページ 118 (Anybusモジュールオブジェクト)
11	Enable CIP forwarding	Get	BOOL	False	値 意味 True 不明なCIPオブジェクトと不明なアセンブリオブジェクトインスタンスに対する要求はアプリケーションへ送られます。 False 不明なCIPオブジェクトと不明なアセンブリオブジェクトインスタンスへの要求はアプリケーションへ送られません。 Process_CIP_Object_Requestに関するコマンド詳細も参照してください。
12	Enable Parameter Object	Get	BOOL	True	値 意味 True CIP/パラメータオブジェクトを有効化 False CIP/パラメータオブジェクトを無効化
13	Input-Only heartbeat instance number	Get	UINT16	0003h	アセンブリオブジェクト (04h), ページ 82 (CIPオブジェクト) の「インスタンス03hアトリビュート (ハートビート、Input-Only)」を参照してください。
14	Listen-Only heartbeat instance number	Get	UINT16	0004h	アセンブリオブジェクト (04h), ページ 82 (CIPオブジェクト) の「インスタンス04hアトリビュート (ハートビート、Listen-Only)」を参照してください。
15	Assembly object Configuration instance number	Get	UINT16	0005h	アセンブリオブジェクト (04h), ページ 82 (CIPオブジェクト) の「インスタンス05hアトリビュート (コンフィグレーションデータ)」を参照してください。
16	Disable Strict IO Match	Get	BOOL	False	Trueであれば、モジュールは設定されたIOサイズ以下のサイズを持つクラス1コネクション要求を受け入れます。
17	Enable unconnected routing	Get	BOOL	False	Trueであれば、モジュールは非コネクション型のCIPルーティングを有効にします。これにより、CIPポートマッピングオブジェクトの内容の初期アップロードもトリガされます。

#	名前	アクセス	データ型	デフォルト値	コメント						
18	Input-Only extended heartbeat instance number	Get	UINT16	0006h	アセンブリオブジェクト (04h), ページ 82 (CIPオブジェクト) の「インスタンス06hアトリビュート (ハートビート、拡張Input-Only)」を参照してください。						
19	Listen-Only extended heartbeat instance number	Get	UINT16	0007h	アセンブリオブジェクト (04h), ページ 82 (CIPオブジェクト) の「インスタンス06hアトリビュート (ハートビート、拡張Listen-Only)」を参照してください。						
20	Interface label port 1	Get	CHARの配列	ポート1	このアトリビュートの値はイーサネットリンクオブジェクト、インスタンス#1のインターフェースラベルを変更するために使用されます。						
21	Interface label port 2	Get	CHARの配列	ポート2	このアトリビュートの値はイーサネットリンクオブジェクト、インスタンス#2のインターフェースラベルを変更するために使用されます。						
22	Interface label internal port	Get	CHARの配列	内部	このアトリビュートの値はイーサネットリンクオブジェクト、インスタンス#3のインターフェースラベルを変更するために使用されます。						
23 ~ 25	(予約)										
26	Enable EtherNet/IP QuickConnect	Get	BOOL	False	<table border="0"> <tr> <td>値</td> <td>意味</td> </tr> <tr> <td>True</td> <td>EtherNet/IP QuickConnect機能が有効。</td> </tr> <tr> <td>False</td> <td>EtherNet/IP QuickConnect機能が無効。</td> </tr> </table> <p>モジュールがEIP QuickConnect機能を使用するように設定されている場合、EDSファイルの変更が必要です。EDSファイルが変更されたら、モジュールのアイデンティティも変更し、モジュールの認証が必要になります。</p>	値	意味	True	EtherNet/IP QuickConnect機能が有効。	False	EtherNet/IP QuickConnect機能が無効。
値	意味										
True	EtherNet/IP QuickConnect機能が有効。										
False	EtherNet/IP QuickConnect機能が無効。										
27 ~ 28	(予約)										
29	Ignore Sequence Count Check	Get	BOOL	False	<p>このアトリビュートを「True」に設定すると、モジュールはconsumedクラス1データに対するシーケンスカウントチェックを無視します。これは、変更されたデータや新たなデータだけではなく、オリジネータから受信されたすべてのデータがアプリケーションにコピーされることを意味します。変更データだけではなくすべてのデータをコピーすることはCIP仕様違反しています。これはモジュールのパフォーマンスにも影響します。</p> <p>このフラグを「True」に設定するときは、十分にご注意ください。HMS Industrial Networksはすべてのデータをコピーしたことによるパフォーマンスへの影響の如何について、パフォーマンス測定を実行せず、保証もいたしません。</p>						
30	ABCC ADI Object Number	Get	UINT16	00A2h	このアトリビュートはADIオブジェクト (CIPオブジェクト) のオブジェクト番号を変更するか、ADIオブジェクト (CIPオブジェクト) を無効にします。有効なオブジェクト番号はベンダー固有範囲 (0064h - 00C7hと0300h - 04FFh)にあります。その他の値はADIオブジェクトを無効にします。						
31	Enable DLR	Get	BOOL	True	<table border="0"> <tr> <td>値</td> <td>意味</td> </tr> <tr> <td>True</td> <td>DLR機能が有効</td> </tr> <tr> <td>False</td> <td>DLR機能が無効</td> </tr> </table>	値	意味	True	DLR機能が有効	False	DLR機能が無効
値	意味										
True	DLR機能が有効										
False	DLR機能が無効										
32	Enable CIP Sync	Get	BOOL	False	<table border="0"> <tr> <td>値</td> <td>意味</td> </tr> <tr> <td>True</td> <td>CIP Sync機能が有効</td> </tr> <tr> <td>False</td> <td>CIP Sync機能が無効</td> </tr> </table>	値	意味	True	CIP Sync機能が有効	False	CIP Sync機能が無効
値	意味										
True	CIP Sync機能が有効										
False	CIP Sync機能が無効										

複数のアセンブリインスタンス

アセンブリマッピングオブジェクトはアプリケーションによって定義されたインスタンスをリストするクラスレベル(ライトPD インスタンスリストとリードPD インスタンスリスト)において二つの配列を持っています。EtherNet/IP ホストオブジェクト (Producingインスタンス番号とConsumingインスタンス番号) のアトリビュート7と8の配列は、アセンブリマッピングオブジェクトにあるインスタンスリストと結合されています。配列はアプリケーションによって定義された各アセンブリインスタンスを表す、対応するCIPインスタンス番号をリストします。

詳細については以下を参照してください。

- [アセンブリマッピングオブジェクト \(EBh\) の使用, ページ 19](#)
- 『Anybus CompactCom 40 Software Design Guide』、「アセンブリマッピングオブジェクト (EBh)」

コマンドの詳細: Process_CIP_Object_Request

カテゴリ

拡張

詳細

コマンドコード: 10h
 有効な対象: オブジェクト

説明

‘Enable CIP Request Forwarding (CIP要求転送有効化)’-アトリビュート (#11)を設定することによって、未実装のCIPオブジェクトと不明なアセンブリオブジェクトインスタンスに対するすべての要求はこのコマンドによってホストアプリケーションへ転送されます。次にアプリケーションは要求を評価し、適正な応答を返す必要があります。モジュールは一つのCIP要求をサポートし、追加の要求があればモジュールによって拒否されます。

ホストインターフェースでのテレグラム長が限られていることから、要求データサイズが255バイトを超えてはならないことに注意してください。これを超えた場合、モジュールは‘resource unavailable (リソース利用不可)’の応答を要求のオリジネータへ送り、メッセージはホストアプリケーションへ転送されません。

このコマンドはAnybus CompactCom 40 DeviceNetにおける ‘Process_CIP_Request’ コマンドに類似していますが同じではありません。

- コマンド詳細

フィールド	内容	コメント
CmdExt[0]	CIPサービスコード	元のCIP要求に基づくCIPサービスコード
CmdExt[1]	要求パスサイズ	要求パスフィールドの16ビットワード長
MsgData[0... m]	要求パス	CIP EPATH (クラス、インスタンス、アトリビュートなど)
MsgData[m... n]	要求データ	サービス固有データ

- 応答詳細

フィールド	内容	コメント
CmdExt[0]	CIPサービスコード	(応答ビットをセット)
CmdExt[1]	00h	(予約、ゼロに設定)
MsgData[0]	ゼネラルステータス	CIPゼネラルステータスコード
MsgData[1]	追加ステータスのサイズ	追加ステータス配列の16ビットワード長
MsgData[2... m]	追加ステータス	追加ステータス (該当する場合)
MsgData[m... n]	応答データ	実際の応答データ (該当する場合)



この機能を使用するときは、ホストアプリケーション ファームウェアにある全オブジェクトについて共通のCIPクラスアトリビュート (アトリビュート #1、Revision) を実装することを忘れないでください。これを怠るとモジュールがコンフォーマンステストに合格できなくなります。

コマンドの詳細: Set_Configuration_Data

カテゴリ

拡張

詳細

コマンドコード: 11h
 有効な対象: オブジェクト

説明

CIP 「Forward_Open」サービスのデータセグメントにコンフィグレーションデータが格納されていると、このコマンドによってこのデータがホストアプリケーションへ転送されます。実装されている場合、ホストアプリケーションは要求を評価して適正な応答を返す必要があります。分割が使用されます。詳細は189ページの“メッセージの分割”をご参照ください。モジュールが受け入れ可能なコンフィグレーションデータの最大合計量は458バイトです。

コンフィグレーションデータをサポートするには、このコマンドが実装されている必要があります。実装されていない場合、「Forward_Open」要求は拒否されます。

- コマンド詳細

フィールド	内容	コメント
CmdExt[0]	-	(予約、無視)
CmdExt[1]	分割制御ビット	メッセージの分割, ページ 142を参照
MsgData[0 - 1]	Producingコネクションポイント	オリジネータによって要求される、Producingコネクションポイント。
MsgData[2 - 3]	Consumingコネクションポイント	オリジネータによって要求される、Consumingコネクションポイント。
MsgData[4... n]	Data	実際のコンフィグレーションデータ

CIP Forward_openサービスがコンフィグレーションデータを含んでいることを受けてSet_Configuration_Dataコマンドがアプリケーションに送信される場合、ProducingコネクションポイントはMsgData[0-1]に、ConsumingコネクションポイントはMsgData[2-3]に表されます。ただし、Set_Configurationコマンドは、CIPアセンブリオブジェクトに対するSet_Attribute_Singleサービスまたは一致しないNULL Forward Openサービス要求を受けて送られることもあります。いずれの場合にもMsgData[0-1]とMsgData[2-3]には0 (ゼロ) が含まれます。

- 応答詳細 (成功)

フィールド	内容	コメント
CmdExt[0]	00h	(予約、ゼロに設定)
CmdExt[1]	00h	(予約、ゼロに設定)

- 応答の詳細 (エラー)

フィールド	内容	コメント
CmdExt[0]	00h	(予約、ゼロに設定)
CmdExt[1]	00h	(予約、ゼロに設定)
MsgData[0]	エラーコード	Anybusエラーコード
MsgData[1]	拡張エラーコード	AnybusエラーコードがFFhに設定されている場合、拡張エラーコードは下表で示すように解釈されなければなりません。
MsgData[2...3]	インデックス	拡張エラーコードが02h (無効なコンフィグレーション)に設定されていると、このパラメータは失敗したアトリビュートへ向けてポイントします。

拡張エラーコード

エラーコードがFFh (オブジェクト固有エラー) に等しい場合、拡張コードは次のように解釈されます。

コード	内容	CIP番号	CIPステータスコード	追加情報
01h	所有権競合	01h	接続失敗	コンフィグレーションデータはforward open要求に提供されました。
		10h	デバイスステートの競合	コンフィグレーションデータはアセンブリオブジェクトへのset要求に渡されました。
02h	無効なコンフィグレーション	09h	不正なアトリビュートデータ	CIP 拡張エラーコード: MsgData[2 - 3]からの値を使用してください。拡張エラーコードは、Forward Open要求から要求が来た場合しか使用できず、明示的なset要求には使用できません。

- [コネクションマネージャ \(06h\)](#), [ページ 85](#) (CIPオブジェクト)
- [メッセージの分割](#)

コマンドの詳細: Process_CIP_Routing_Request

カテゴリ

拡張

詳細

コマンドコード: 12h
 有効な対象: オブジェクト

説明

モジュールは「Unconnected_Send」サービスの中の最初のパスを切り取り、ルーティングの継続が可能か否かを評価します (要求されたポートがポートオブジェクトの内部に存在するか否かの確認など)。切り取られたパスが最終パスであった場合、アプリケーションへ渡されるコンテンツはターゲットノードへ送られるCIP要求となります。それ以外の場合は、ルートパス情報が更新された「Unconnected_Send」サービスとなります。

モジュールは一個の保留要求のみサポートします。追加の要求はモジュールによって拒否されます。

ホストインターフェースでのテレグラム長が限られているため、データ長が255バイトを超えてはならないことに注意してください。超えた場合、モジュールはオリジネータの要求を拒否し (「Resource unavailable」)、このコマンドはホストアプリケーションへ発行されません。

- コマンド詳細

フィールド	内容	コメント
CmdExt[0]	-	(予約、無視)
CmdExt[1]	-	(予約、無視)
MsgData[0... n]	ターゲットパス	EPathとしてエンコードされたターゲットパス。
MsgData[n+1]	Time_tick	タイムアウトパラメータの更新後に有効
MsgData[n+2]	Time-out_ticks	タイムアウトパラメータの更新後に有効
MsgData[n+3... m]	CIPメッセージ	ルートするCIPメッセージ

- 応答詳細

フィールド	内容	コメント
CmdExt[0]	00h	(予約、ゼロに設定)
CmdExt[1]	00h	(予約、ゼロに設定)
MsgData[0]	CIPサービス	実際のCIPサービスコード、応答ビットをセット
MsgData[1]	00h	(予約、ゼロに設定)
MsgData[2]	ゼネラルステータス	実際のCIPゼネラルステータスコード
MsgData[3]	追加ステータスのサイズ	追加ステータス配列の16ビットワード長
MsgData[4... n]	追加ステータス配列	追加ステータス (該当する場合)
MsgData[n+1... m]	応答データ	実際の応答データ

下記も参照してください。

- [ポートオブジェクト \(F4h\), ページ 104](#) (CIPオブジェクト)
- [CIPポートコンフィグレーションオブジェクト \(0Dh\), ページ 152](#)

コマンドの詳細: Get_Configuration_Data

カテゴリ

拡張

詳細

コマンドコード: 13h
 有効な対象: オブジェクト

説明

コンフィグレーションデータがネットワークから要求された場合、Anybusはこのコマンドをアプリケーションに対して発行します。アプリケーションは保存されたコンフィグレーションデータ応答メッセージを送ります。

ホストインターフェースでのテレグラム長が限られており、分割が使用されます。モジュールが受け入れ可能なコンフィグレーションデータの最大合計量は458バイトです。

このコマンドはコンフィグレーションデータに対応するために実装されなければなりません。実装されない場合、要求はAnybus モジュールによって拒否されます。

- コマンド詳細

フィールド	内容	コメント
CmdExt[0]	00h	-
CmdExt[1]	00h	-
MsgData[0... n]	-	拡張メッセージデータなし

- 応答詳細 (成功)

フィールド	内容	コメント
CmdExt[0]	00h	(予約、ゼロに設定)
CmdExt[1]	分割制御ビット	メッセージの分割, ページ 142 を参照
MsgData[0 - n]	状態	アプリケーションからのコンフィグレーションデータ

- 応答の詳細 (エラー)

フィールド	内容	コメント
CmdExt[0]	00h	(予約、ゼロに設定)
CmdExt[1]	分割制御ビット	メッセージの分割, ページ 142 を参照
MsgData[0]	状態	Anybus プロトコルエラーコード

12.9 Ethernet ホストオブジェクト (F9h)

オブジェクトの説明

このオブジェクトはEthernetの機能をホストアプリケーションに実装します。

サポートされているコマンド

オブジェクト : Get_Attribute
 インスタンス : Get_Attribute
 Set_Attribute

オブジェクトアトリビュート (インスタンス#0)

#	名前	アクセス	データ型	値
1	Name	Get	CHARの配列	「イーサネット」
2	Revision	Get	UINT8	02h
3	Number of instances	Get	UINT16	0001h
4	Highest instance no.	Get	UINT16	0001h

インスタンスアトリビュート (インスタンス#1)

- アトリビュートが実装されない場合、デフォルト値が使用されます。
- モジュールは有効なMACアドレスがプレプログラムされています。そのアドレスを使用するには、アトリビュート #1を実装しないでください。
- モジュールがプレプログラムされたMACアドレスを使用する場合、PROFINETデバイス専用のアトリビュート#9と#10は実装しないでください。
- 新たなMACアドレスがPROFINETデバイスに割り当てられるとこれらのアドレス (アトリビュート #1、#9、#10) は連続していなければなりません。例 (xx:yy:zz:aa:bb:01)、(xx:yy:zz:aa:bb:02)、(xx:yy:zz:aa:bb:03) では先頭のオクテット5個が不変です。

#	名前	アクセス	データ型	デフォルト値	コメント
1	MAC address	Get	UINT8の配列	-	6 バイトの物理アドレス値。プレプログラムされたMacアドレスをオーバーライドします。新たなMacアドレス値をIEEEから取得しなければならないことにご注意ください。 プレプログラムMacアドレスを使用する場合はこのアトリビュートを実装しないでください。
2	Enable HICP	Get	BOOL	True (有効)	HICPの有効化/無効化
3	Enable Web Server	Get	BOOL	True (有効)	Webサーバの有効/無効 (トランスペアレントイーサネットが有効化されている場合は不使用。)
4	(予約)				Anybus CompactCom 30 アプリケーション用に予約済み。
5	Enable Web ADI access	Get	BOOL	True (有効)	Web ADIアクセスの有効/無効 (トランスペアレントイーサネットが有効化されている場合は不使用。)
6	Enable FTP server	Get	BOOL	True (有効)	FTPサーバの有効化/無効化 (トランスペアレントイーサネットが有効になっている場合、またはデバイスがIoTセキュア機能をサポートしている場合は不使用。)
7	Enable admin mode	Get	BOOL	False (無効)	Adminモードの有効化/無効化 (トランスペアレントイーサネットが有効化されている場合は不使用。)
8	Network Status	Set	UINT16	-	下記を参照してください。

#	名前	アクセス	データ型	デフォルト値	コメント
9	Port 1 MAC address	Get	UINT8の配列	-	注: このアトリビュートはPROFINETデバイス用のみに有効です。ポート1用の6バイトMACアドレス(LLDPプロトコルでは必須)。この設定はホストPROFINET IOオブジェクトにおいてどのポートMACアドレスをもオーバーライドします。プレプログラムMacアドレスを使用する場合はこのアトリビュートを実装しないでください。
10	Port 2 MAC address	Get	UINT8の配列	-	注: このアトリビュートはPROFINETデバイス用のみに有効です。ポート2用の6バイトMACアドレス(LLDPプロトコルでは必須)。この設定はホストPROFINET IOオブジェクトにおいてどのポートMACアドレスをもオーバーライドします。プレプログラムMacアドレスを使用する場合はこのアトリビュートを実装しないでください。
11	Enable ACD	Get	BOOL	True (有効)	ACDプロトコルの有効/無効。 ACD機能がこのアトリビュートによって無効な場合、CIP TCP/IPオブジェクト(F5h)のACDアトリビュートは利用できません。
12	Port 1 State	Get	ENUM	0 (有効)	Ethernet ポート1の状態。 <ul style="list-style-type: none"> このアトリビュートは、EtherCATおよびイーサネットPOWERLINKデバイスではポート1が常時有効になっているため、読み取られません。 <p>00h: 有効</p> <p>01h: 無効。 ポートは存在しているとして扱われます。ポートへの参照は存在可能です(ネットワークプロトコルやWebサイトなど)。</p>
13	Port 2 State	Get	ENUM	0 (有効)	Ethernetポート2の状態。 <ul style="list-style-type: none"> このアトリビュートは、EtherCATおよびイーサネットPOWERLINKデバイスではポート2が常時有効になっているため、読み取られません。 <p>00h: 有効</p> <p>01h: 無効。 ポートは存在しているとして扱われます。ポートへの参照は存在可能です(ネットワークプロトコルやWebサイトなど)。</p> <p>02h: 非アクティブ。 このアトリビュートは、物理ポートが1つしかないデバイスではこの値に固定されています。ポート2の全ての機能は無効です。このポートを参照することはできません。 注: この機能はPROFINET、EtherNet/IP、Modbus-TCPデバイスで利用可能です。</p>
14	(予約)				
15	Enable reset from HICP	Get	BOOL	0 = False	モジュールをHICPからリセットするオプションを有効化します。
16	IP configuration	Set	構造体: UINT32 (IPアドレス) UINT32 (サブネットマスク) UINT32 (ゲートウェイ)	N/A	コンフィグレーションが割り当てられるか変更される場合は常に、Anybus CompactComモジュールがこのアトリビュートを更新します。

#	名前	アクセス	データ型	デフォルト値	コメント
17	IP address byte 0-2	Get	UINT8[3]の配列	[0] = 192 [1] = 168 [2] = 0	IPアドレスの先頭3バイト。コンフィギュレーションスイッチ値が1~245に設定されている場合、スタンダロンシフトレジスタモードで使用されます。この場合、IPアドレスは以下のようになります。 Y[0].Y[1].Y[2].X Y0-2はこのアトリビュートによって設定され、最後のバイトXはコンフィギュレーションスイッチによって設定されます。
18	Ethernet PHY Configuration	Get	BITS16の配列	各ポートで 0x0000	Ethernet PHYコンフィギュレーションのビットフィールド。配列の長さは製品のEthernetポート数と等しくなければなりません。各要素は1個のEthernetポート(要素 #0はEthernetポート#1に、要素#1はEthernetポート#2にマップなど)のコンフィギュレーションを表します。 注: EtherNet/IPとModbus-TCPデバイスでのみ有効です。 ビット0: オートネゴシエーション失敗時のduplex 0 = 半二重 1 = 全二重 ビット1-15: 予約
20	SNMP read-only community string	Get	CHARの配列	「public」	注: このアトリビュートはPROFINETデバイス用のみに有効です。SNMP読み取り専用コミュニティ文字列を設定します。最大長は32です。
21	SNMP read-write community string	Get	CHARの配列	「private」	注: このアトリビュートはPROFINETデバイス用のみに有効です。SNMP読み取り-書き込みコミュニティ文字列を設定します。最大長は32です。
22	DHCP Option 61 source	Get	ENUM	0 (無効)	注: このアトリビュートは、現在のところEtherNet/IPデバイスにのみ有効です。 以下を参照してください (DHCP Option 61、Client Identifier)
23	DHCP Option 61 generic string	Get	UINT8の配列	N/A	注: このアトリビュートは、現在のところEtherNet/IPデバイスにのみ有効です。 以下を参照してください (DHCP Option 61、Client Identifier)
24	Enable DHCP Client	Get	BOOL	1 = True	注: このアトリビュートは、現在のところEtherNet/IPおよびPROFINETデバイスで有効です。 DHCPクライアント機能の有効化/無効化 0: DHCPクライアント機能が無効 1: DHCPクライアント機能が有効
25	Enable WebDAV Server	Get	BOOL	1 = True	注: このアトリビュートは、現在のところIIoTセキュア機能を持つデバイスで有効です。 WebDAVサーバの有効化/無効化 0: WebDAV機能が無効 1: WebDAV機能が有効

Network Status

このアトリビュートは、次のようにネットワーク全体のステータスを示すビットフィールドを保持します。

ビット	内容	説明	コメント
0	Link	グローバルリンクの現在の状態 1 = リンク検出 0 = リンクなし	
1	IP established	1 = IPアドレス確立 0 = IPアドレス未確立	
2	(予約)	(マスクオフで無視)	
3	Link port 1	ポート1リンクの現在の状態 1 = リンク検出 0 = リンクなし	EtherCATのみ: このリンク状態は、Anybus CompactComが ⁶ Ethernet over EtherCAT (EoE) を使用して通信できるかどうかを示します。つまり、論理的なEoEポートリンクのステータスを示し、物理的なEtherCATポートのリンクステータスとは関係ありません。
4	Link port 2	ポート2リンクの現在の状態 1 = リンク検出 0 = リンクなし	EtherCATでは不使用
5...15	(予約)	(マスクオフして無視する)	

DHCP Option 61 (Client Identifier)



EtherNet/IPデバイスにのみ有効

DHCP Option 61 (Client Identifier) を使用すると、エンドユーザーはDHCPドメイン内で一意である必要のある、一意の識別子を指定することができます。

Client Identifierのソースの設定には、アトリビュート#22 (DHCP Option 61のソース) が使用されます。以下の表は、さまざまなソースに対するClient IDの定義とその説明を示しています。

値	ソース	説明
0	Disable	DHCP Option 61は無効です。このアトリビュートがアプリケーションに実装されていない場合は、これがデフォルト値となります。
1	MACID	MACIDがClient Identifierとして使用されます
2	Host Name	設定されたホスト名がClient Identifierとして使用されます
3	Generic String	アトリビュート#23がClient Identifierとして使用されます

アトリビュート#22が3 (Generic String) に設定されているときは、アトリビュート#23 (DHCP Option 61 generic string) はClient Identifierを設定するために使用されます。アトリビュート#23はTypeフィールドとClient Identifierを含み、RFC 2132の定義に準拠するものとします。アトリビュート#23によってモジュールに渡すことのできる許容最大長は64オクテットです。

例:

アトリビュート#22が3 (Generic String) に設定されていて、アトリビュート#23に0x01、0x00、0x30、0x11、0x33、0x44、0x55が含まれている場合、Client IdentifierはMACID 00:30:11:33:44:55を持つイーサネットメディア型として表されます。

例 2 :

アトリビュート#22 が2 (ホスト名) に設定されている場合、アトリビュート#23は無視されてClient Identifierは設定されたホスト名と同一になります。

このページは意図的に空白になっています

A 機能の分類

Anybus CompactComとアプリケーションの属性やサービスを含むオブジェクトは二種類に分かれます：基本と拡張。

A.1 基本

このカテゴリには、実装または使用しなければならない必須のオブジェクト、属性、サービスが含まれます。Anybus CompactComを起動し、選択したネットワークプロトコルでデータを送受信するにはこのカテゴリで十分であり、産業用ネットワークの基本機能が使用されます。

製品認証を可能にする追加オブジェクトなどもここに分類されます。

A.2 拡張

このカテゴリのオブジェクトを使用すると、アプリケーションの機能を拡張できます。ネットワークにおける基本的なデータ交換だけでなく、産業用ネットワーク固有の機能を利用できるようになります。これにより、アプリケーションの価値が高まります。

一部の機能は非常に高度であったり、使用されることが稀であったりすることがあります。大半の利用可能なネットワーク機能は有効でありアクセスできるので、産業ネットワーク仕様にアクセスできることが必要な場合があります。

B 実装詳細

B.1 SUPビットの定義

監視ビット (SUP) は、他のネットワーク機器によってネットワークへの参加が管理されていることを表します。EtherNet/IPの場合は、SUPビットがセットされていることは、1つまたは複数のCIP (クラス1またはクラス3)コネクションがモジュールに向かってオープンされていることを意味します。

B.2 Anybusステートマシン

下表にAnybusステートマシンとEtherNet/IPネットワークの関連性について表します

Anybusの状態	実装	コメント
WAIT_PROCESS	モジュールはクラス1コネクションがオープンされるまでこのステートに留まります。	-
ERROR	クラス1コネクションエラー 重複IPアドレスが検出された	-
PROCESS_ACTIVE	エラーがないクラス1コネクションが有効 (Exclusive-Ownerコネクションの32ビットのRun/IdleヘッダーのRUNビットはセットされている)	消費コネクションのみで有効。
IDLE	クラス1コネクションがアイドル。	
EXCEPTION	予想されないエラー、例えばウォッチドッグのタイムアウトなど。	MS LEDが赤点灯(重大な障害を示す) NS LEDが消灯

B.3 Application Watchdog Timeout Handling

アプリケーションウォッチドッグのタイムアウトが検出されると即座に、モジュールはネットワークから切断され、EXCEPTIONステートに変わります。その他のネットワーク固有の動作は実行されません。

C セキュアHICP (セキュアホストIPコンフィグレーション プロトコル)

C.1 概要

Anybus CompactCom 40 EtherNet/IPは、Anybus IPconfigユーティリティが設定変更のために使用するセキュアHICPプロトコル (IPアドレス、サブネットマスク、DHCPの有効化/無効化など) をサポートします。Anybus IPconfigはHMSのウェブサイト www.anybus.com から無償でダウンロード可能です。このユーティリティでネットワークにUDPポート3250から接続されたAnybusのどの製品のネットワーク設定にもアクセスできます。

プロトコルはデバイスの再起動のためにセキュアな認証と能力を提供します。

C.2 操作

アプリケーションが起動すると、ネットワークはAnybus製品を自動的にスキャンします。**Scan**をクリックすると、随時ネットワークを再スキャンできます。

モジュールのネットワーク設定を変更するには、リストのエントリをダブルクリックします。モジュールの設定を含むウィンドウが表示されます。

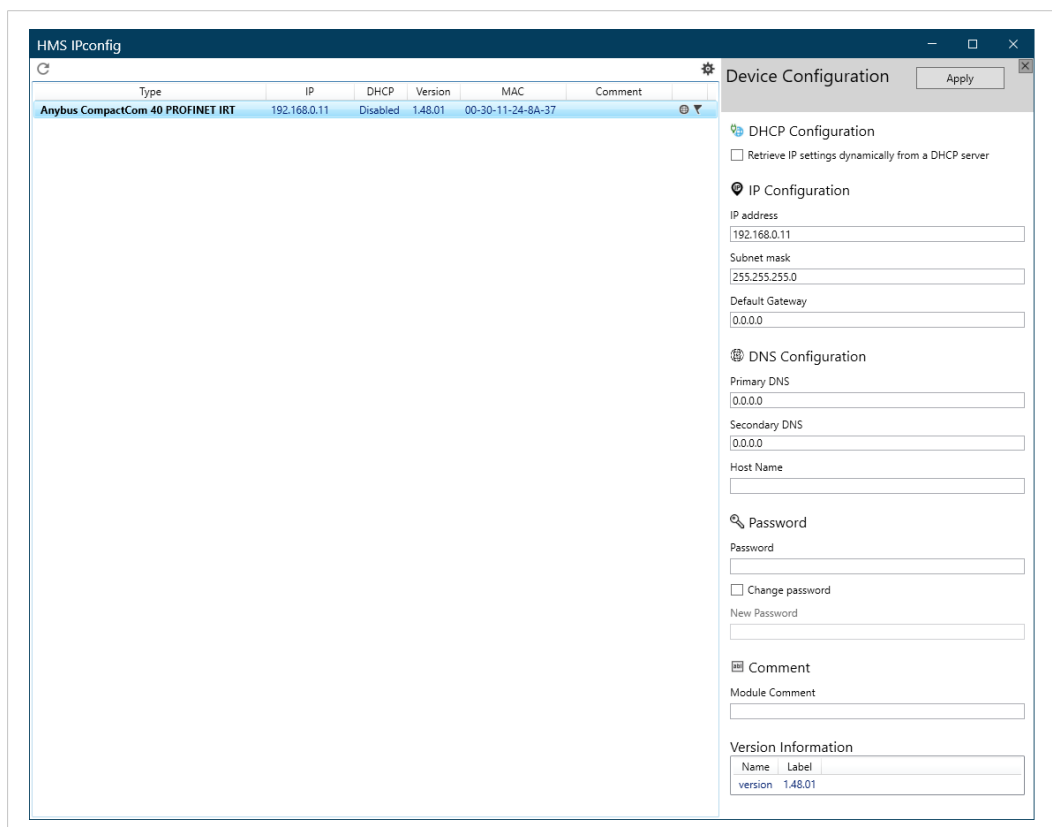


図 8

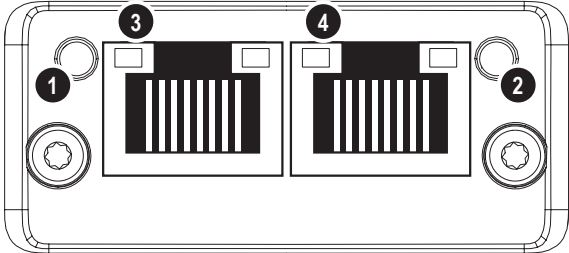
Setをクリックして新たな設定を適用するか、**Cancel**をクリックしてすべての変更を取り消します。オプションとして、パスワードを使用すると許可されていないアクセスからコンフィグレーションを保護することができます。

D 技術仕様

D.1 前面図

D.1.1 前面図 (Ethernetコネクタ)

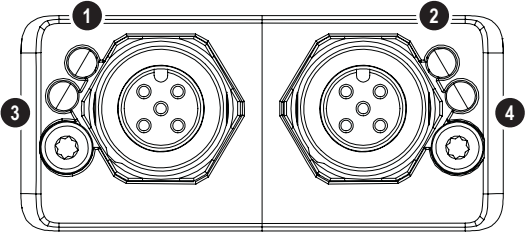
#	項目	コネクタ
1	ネットワークステータスLED	イーサネット、 RJ45
2	モジュールステータスLED	
3	リンク/アクティビティLED (ポート1)	
4	リンク/アクティビティLED (ポート2)	



起動中に、ネットワークステータスLEDとモジュールステータスLEDに対して一連のテストが実行されます。

D.1.2 前面図 (M12コネクタ)

#	項目	コネクタ
1	ネットワークステータスLED	M12
2	モジュールステータスLED	
3	リンク/アクティビティLED (ポート1)	
4	リンク/アクティビティLED (ポート2)	



起動中に、ネットワークとモジュールステータスLEDに対してテスト手順が実行されます。

D.1.3 ネットワークステータスLED

LEDの状態	説明
オフ	電力供給なしまたはIPアドレスが設定されていない
緑	オンライン、1つまたは複数のコネクションが確立した (CIP Class 1または3)
緑点滅	オンライン、コネクションが確立していない
赤	IPアドレスの重複、致命的エラー
赤点滅	1つまたは複数のコネクションがタイムアウトした (CIP Class 1または3)

D.1.4 モジュールステータスLED

LEDの状態	説明
オフ	電力供給なし
緑	Run 状態のスカナによって制御されている状態、かつ、CIP Syncが有効になっている場合は時間がグラウンドマスタークロックに同期している
緑点滅	ネットワーク設定が未設定、スカナがアイドル状態、または、CIP Syncが有効になっている場合は時間がグラウンドマスタークロックに同期している
赤	重大な障害 (EXCEPTION状態、致命的エラーなど)
赤点滅	復旧可能な障害。モジュールは設定されているが、保存されたパラメータが現在使用中のパラメータと異なる。

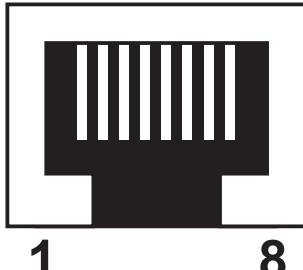
D.1.5 リンク/アクティビティ (LED 3/4)

LEDの状態	説明
オフ	リンクなし、動作していない
緑	リンク (100 Mbit/s) 確立済み
緑、はやい点滅	動作 (100 Mbit/s)
黄色	リンク (10 Mbit/s) 確立済み
黄色、はやい点滅	動作 (10 Mbit/s)

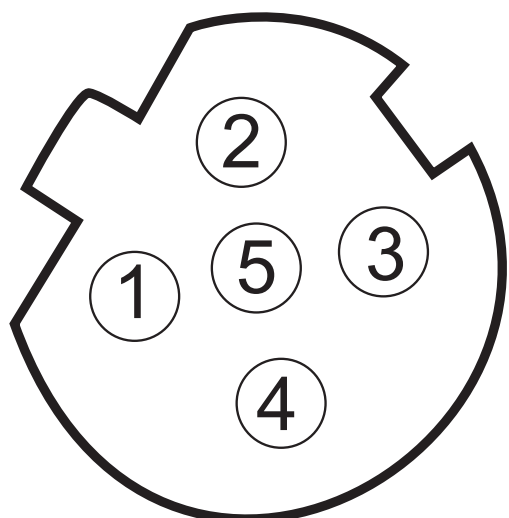
D.1.6 Ethernet インターフェース

Ethernet インターフェース (RJ45 コネクタ)

Ethernet インターフェース10/100Mbit、全二重または半二重通信。

ピン番号	説明	
4、5、7、8	シリアルRC回路を経由してシャーシアースに接続	
6	RD-	
3	RD+	
2	TD-	
1	TD+	
ハウジング	ケーブルシールド	

D.1.7 M12コネクタ、コードD

ピン	Name	説明	
1	TXD+	正の送信	
2	RXD+	正の受信	
3	TXD-	負の送信	
4	RXD-	負の受信	
5 (スレッド)	シールド	シールド	

D.2 ファンクショナルアース (FE) の要件

適正なEMC動作を保証するため、モジュールをAnybus CompactCom 40 Hardware Design Guideに説明されているファンクショナルアースにFE pad/FE機構によって正しく接続してください。FE要件に準拠していない限り適正なEMC動作は保証されません。

D.3 電源

D.3.1 電源電圧

Anybus CompactCom 40 EtherNet/IPには全般的な*Anybus CompactCom 40 Hardware Design Guide*で指定された3.3 Vの安定化電源が必要です。

D.3.2 消費電力

Anybus CompactCom 40 EtherNet/IPはクラスBモジュールの要求事項を満たすように設計されています。現在のハードウェア設計は360 mAまで消費します

製品開発に継続的に取り組むというHMSのポリシーに従って、本製品の消費電力に関する厳密な要件を予告なく変更する権利を留保します。ただし、いかなる場合でもAnybus CompactCom 40 EtherNet/IPはクラスBモジュールに留まります。



ホストアプリケーションにおける電源の設計は、ある1つの製品の消費電力に関する厳密な要件ではなく、全般的な『*Anybus CompactCom Hardware Design Guide*』で記述された消費電力の分類に基づいて行うことを強く推奨します。

D.4 環境仕様

詳細については*Anybus CompactCom 40 Hardware Design Guide*を参照してください。

D.5 EMC準拠

詳細については*Anybus CompactCom 40 Hardware Design Guide*を参照してください。

E タイミングと性能

E.1 基本情報

この章には、Anybus CompactCom 40 EtherNet/IPについて検証され文書化されたタイミングと性能パラメータが記載されています。

カテゴリ	パラメータ	ページ
起動時の遅延	T1, T2	195
NW_INITの処理	T100	195
イベントベースのWrMsgビジー時間	T103	195
イベントベースのプロセスデータ遅延時間	T101, T102	195

詳細については、『Anybus CompactCom 40 Software Design Guide』を参照してください。

E.2 内部タイミング

E.2.1 起動時の遅延

以下のパラメータは、/RESETが解放されてから指定のイベントが発生するまでの時間として定義されています。

パラメータ	説明	最大	単位
T1	Anybus CompactCom 40 EtherNet/IPモジュールが最初のアプリケーション割り込みを生成する(パラレルモード)	64	ms
T2	Anybus CompactCom 40 EtherNet/IPモジュールが最初のアプリケーションテレグラムを受信・処理できる(シリアルモード)	64	ms

E.2.2 NW_INITの処理

このテストは、Anybus CompactCom 40 EtherNet/IPモジュールがNW_INIT状態で必要な動作を実行するために要する時間を測定します。

パラメータ	条件
ネットワーク固有のコマンドの数。	最大
各方向における、プロセスデータにマッピングされたADI (UINT8 × 1) の数。(ネットワーク固有の最大値がこの規定値より小さい場合、ネットワーク固有の値が使用されます。)	32
イベントベースで処理する場合の、アプリケーションのメッセージ応答時間。	> 1 ms
ピンポン方式で処理する場合の、アプリケーションの応答時間。	> 10 ms
アプリケーションが同時に処理可能な、Anybusの未処理コマンドの数。	1

パラメータ	説明	通信	最大	単位
T100	NW_INITの処理	イベントベースのモード	58	ms

E.2.3 イベントベースのWrMsgビジー時間

アプリケーションがメッセージをポストしてから、モジュールがH_WRMSG領域をアプリケーションに返すまでに要する時間として、イベントベースのWrMsgビジー時間が定義されています。

パラメータ	説明	最小	最大	単位
T103	H_WRMSG領域のビジー時間	6	9	μs

E.2.4 イベントベースのプロセスデータ遅延時間

「リードプロセスデータ遅延時間」は、ネットワークフレームの最後のビットがネットワークインターフェースによって受信されてから、アプリケーションに対してRDPDI割り込みがアサートされるまでの時間として定義されます。

「ライトプロセスデータ遅延時間」は、アプリケーションがライトプロセスデータバッファを交換してから、新規プロセスデータフレームの最初のビットがネットワークに送り出されるまでの時間として定義されます。

これらのテストは16ビットパラレルイベントモードで、新規プロセスデータイベントに対してのみ割り込みをトリガーするという条件で実施されました。8通りのIOサイズ (2、16、32、64、128、256、512、1024バイト) でテストを行い、すべてのIOサイズで同じテスト結果が得られました。

PHY回路によって増加した遅延は、プロセスデータ遅延時間全体に比べるとわずかな量なので、結果に含まれていません。

パラメータ	説明	遅延 (最小)	遅延 (標準値)	遅延 (最大)	単位
T101	リードプロセスデータ遅延時間 IOサイズ32バイトで測定	-	-	84	μs
T102	ライトプロセスデータ遅延時間 IOサイズ32バイトで測定	-	-	106	μs

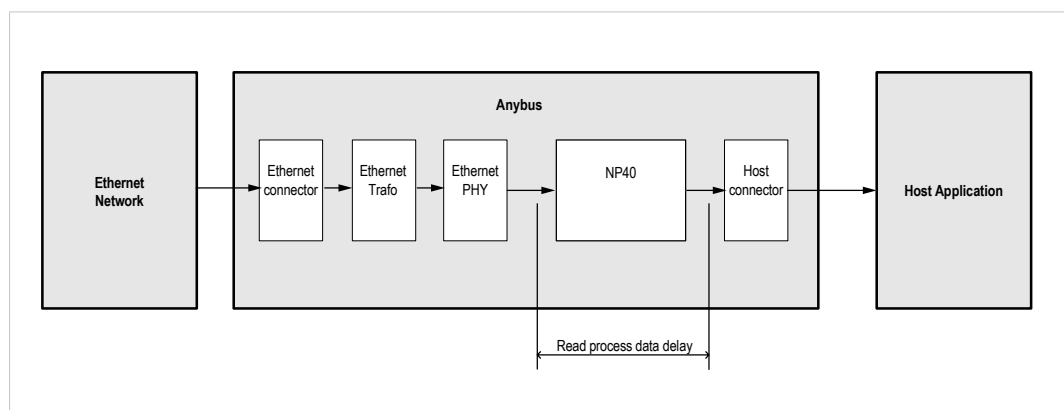


図 9

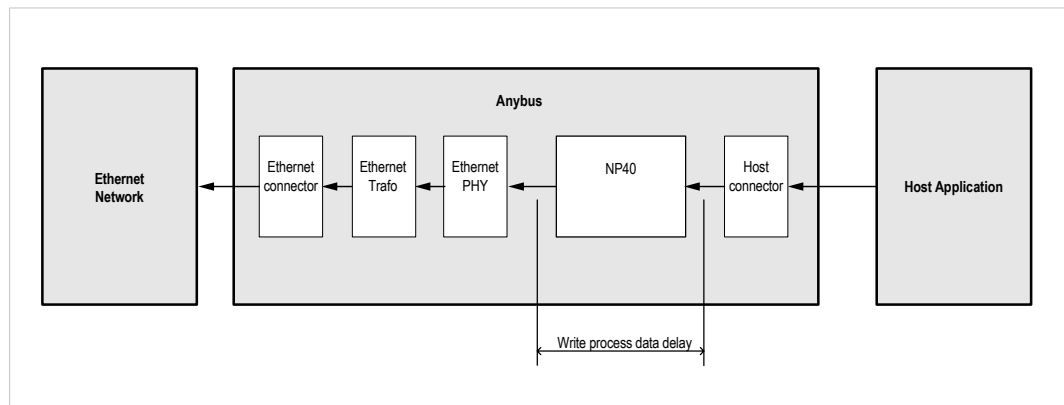


図 10

F コンフォーマンステストガイド

F.1 概要

すべてのパラメータでデフォルト設定を使用する場合、Anybus CompactCom 40 EtherNet/IPはネットワークコンプライアンスが事前に認証されたこととなります。この事前認証は、最終製品が確実に認定されるようにするために行われます。

特定の製品内でEtherNet/IPを使用するには、ベンダーは、ライセンスを受けたEtherNet/IPベンダーとして独自のベンダーIDを持つ必要があります。ベンダーIDの取得についてはwww.odva.orgまでお問い合わせください。

HMS Industrial Networksによって供給されるEDSサンプルファイルのパラメータを変更する場合、新しい認証が必要となります。この章では、Anybus CompactCom 40 EtherNet/IPを使用した貴社製品が、ODVAの定めるネットワーク認証要件に準拠してコンフォーマンステストに合格するためのガイドを示します。

この付録で説明するアクションは一般的な認証プロセスですので、実際にはお使いのデバイスに合わせて製品IDなどを変更してください。



この付録では、コンフォーマンステストの実施と認証に必要な事項についてのガイドラインと例を示します。お使いのアプリケーションの機能によっては、記載以外の手順の実行が必要となる場合があります。

このドキュメント内のすべてのスクリーンショットは、ODVA Conformance Test Software Tool for EtherNet/IP CT14, ©から引用したものです。同ソフトウェアはODVAウェブサイト注文できます。コンフォーマンステストを受けるために製品を提出する前に、このソフトウェアで事前テストを実施する必要があります。

また、コンフォーマンステストを受けるために製品を提出する前に、EtherNet/IPアプリケーションについて記述したコンフォーマンス宣言ファイル (STCファイル) を準備しておく必要があります。

F.2 推奨のテストツール

F.2.1 Wireshark

この無料のオープンソースツールは、ネットワークのキャプチャおよび分析用の事実上の標準ツールとなっています。ODVA TSP、HMS Industrial Networks、幅広いEtherNet/IPユーザーによって非常によく使用されています。Wireshark (www.wireshark.org) はコンピュータのネットワークインターフェースカードを使用してイーサネットトラフィックをキャプチャし、直感的な方法でその内容を表示して、パケットの詳細な分析を可能にします。HMS Industrial Networksの開発者がEtherNet/IPのディセクタ (分析エンジン) に貢献した結果、ユーザーが自社アプリケーションデータのディセクタを独自に作成できるようになっています。Wiresharkの使用についてはドキュメントが豊富にあります。EtherNet/IPテストで重要な情報を入手するのに役立つヒントがいくつかあります。

- 表示フィルタ「CIP」を使用すると、EtherNet/IPトラフィックのみを見ることができます。
- HMS MAC IDによるフィルタ処理が可能です。ソースまたは宛先を「eth.addr[0:3] == 00:30:11」とすると、HMSデバイスとやりとりされるイーサネットメッセージのみが表示されます。
- WiresharkのWebページには、他にも便利なフィルタが多数あります。

F.2.2 NMAP

NMAPはネットワーク検出とセキュリティテストのための無料のオープンソースツールです。NMAPは、どのTCPポートとUDPポートが開いているか、または応答しているかを検出します。また、どのレイヤー3サービスがデバイスでサポートされているかも定義します。ODVAは、オープンポートと必須のレイヤー3サービスに関して厳格なガイドラインを定めています。TSPで使用するNMAPの手順については、ODVAのコンフォーマンステストソフトウェアに付属しているサンプルテストレポートを参照してください。

F.2.3 ODVAコンフォーマンステストソフトウェア

この自動テストソフトウェアは、デバイスのソフトウェア欠陥を確認・誘発・検出するために設計されています。ODVAは、コンフォーマンステストの準備ができるように、ベンダーにこのソフトウェアを年間サブスクリプションという形で販売しています。このソフトウェアは、適合宣言 (STC) ファイルの修正

や作成を行うための最良の方法でもあります。CTRL+Dキーを押すと、STCファイルのデータセクション用GUIが表示されます。

テストの開始

インストールが完了すると、デフォルトのブラウザにWebページが表示されます。このページには、テストソフトウェアの概要と関連ドキュメントの一覧、ならびに各関連ドキュメントの簡単な要約が表示されます。テストのセットアップは、「コンFORMANCEテストソフトウェア・ユーザーマニュアル」に記載されています。

章	内容
1	システム要件とインストール
2	デバイスの選択方法、コンFORMANCE宣言ファイルの修正方法
3	テスト開始前のネットワークセットアップ方法
4	テストソフトウェアの実行方法

ユーザーマニュアル - 重要なポイント

テストソフトウェアを完全に理解するため、コンFORMANCEテストソフトウェア・ユーザーマニュアルを最後までお読みになることを強くお勧めします。このドキュメントで特に重要な部分を以下に記します。

- テストに使用するネットワークインターフェースは、[Setup] メニューに示される利用可能なネットワークインターフェースカードから選択する必要があります。

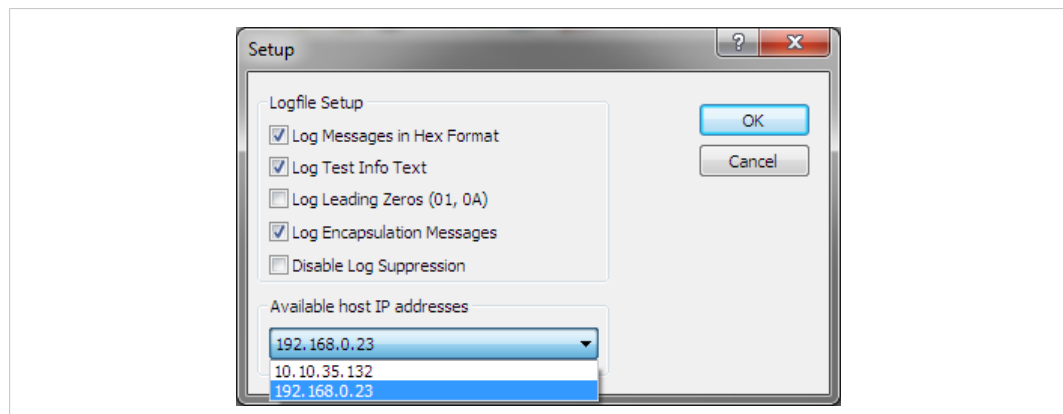


図 11

- ほとんどのデバイスはデフォルトのタイムアウト設定に準拠しますが、デバイスによっては応答基準を緩くする必要があります。この設定は [Set Message Wait Timers] メニューで行えます。
- 最新バージョンのCTテストソフトウェアでは、ネットワークインターフェースカードの2つ目のIPアドレスをユーザーが割り当てる必要があります。
- CTテストのカプセル化ロギング機能を有効にすると、WiresharkのキャプチャとコンFORMANCEテストログを効率的に実行させることができます。

F.2.4 EZ-EDS

EZ-EDSはODVAから入手できる無料のユーティリティです。このツールは電子データシートの編集と確認に非常に役立ちます。電子データシートは、データ編成、コンフィグレーション、パフォーマンス機能を記述するASCII形式のファイルです。通常EDSファイルと呼ばれ、.edsという拡張子を持ちます。EDSファイルはテキストエディタを使用して構築・変更できる一方、EZ-EDSはグラフィカルなユーザーインターフェースを提供することで主要フィールドに注意が向くようになっています。EZ-EDSは、EDSファイルの形式が正しいか否かもテストします。EDSファイルに記入可能な内容の多くはオプションであり、ODVAテストでは内容ではなく正しい形式が重視されます。

F.2.5 Anybus EDS Generator

Anybus EDS Generatorツールは、Anybus CompactCom 40 EtherNet/IPを使用してデバイスをスキャンすることで、EDSファイルを自動的に生成します。このツールは簡単に使い、お使いの製品に合うように正しく構成されたEDSファイルを提供します。ただし、EZ-EDSでEDSファイルを検証する必要があります。

このツールはHMS Industrial Networks Webサイトで入手できます。

F.2.6 サンプルテストレポート

コンフォーマンステストソフトウェアのサブスクリプションには、EtherNet/IPサンプル テストレポートドキュメントが含まれます。このドキュメントには、自動テストソフトウェアの実行手順のほか、テスト担当者が手動で実行する手順の概要が記載されています。

手動テストの手順

EtherNet/IPの一部の機能は、自動テストソフトウェアでは正しく検証できない場合や、完全に自動化されたテストが開発されたとしてもその適用が現実的ではない場合があります。このような機能については、手動テストの手順とテストの合格条件が、サンプルテストレポートに一覧として挙げられています。これが、テストサービスプロバイダーによって使用される正確な手順と条件となります。

手動でテストする必要がある機能の大半はAnybus CompactComのシステムによって提供され、ホストアプリケーションとのやりとりはありません。したがって、Anybus CompactComを使用している場合はこの長々とした手順を省略して構いませんが、以下の点は確認する必要があります。

- HMSでは、全ての方に、サンプルテストレポートの物理層セクションとEDSテストセクションを完了することをお勧めしています。これにより、LEDの生成ラベリングが正しいことと、コンフォーマンステストを受けるために製品をTSPIに提出する前にEDSファイルの検証が行われることが保証されます。
- DLRが製品で有効になっている場合、すべてのイーサネットポートの速度と双方向性を何らかの方法で設定する必要があります。ホストアプリケーションは以下の方法によって、Anybus CompactComの速度と双方向性を設定するための標準的な手段を無効にすることができます。
 - EtherNet/IPホストオブジェクトのインスタンスアトリビュート#9 (Enable Communication Settings from NET) をFalseに設定することで、イーサネットリンクオブジェクトへのSetアクセスを無効にする。
 - Webサーバを無効にする。Anybus CompactComの標準的なWebページでは、イーサネットポートの速度と双方向性を設定することができます。トランスペアレントイーサネット機能を使用するアプリケーションの場合、Webサーバは常に無効となります。

上記のいずれかの方法による速度と双方向性の設定が不可能な場合、これらを設定するための別の方法がホストアプリケーションから提供されなければなりません。例として、設定に使用できるキーパッドインターフェースをアプリケーションに持たせるという方法があります。

- ホストアプリケーションにIPアドレス設定用のハードウェアスイッチ (DIPスイッチ、ロータリースイッチなど) がある場合や、ACDまたはDLRが無効にされている場合、HMS Industrial NetworksはサンプルテストレポートのTCP/IP Interface Object Tests (セクション4)、Ethernet Link Object Tests (セクション5)、Address Conflict Detection (ACD) Tests (セクション10) に記載されている手順に従って手動テストを行うことを推奨します。

F.3 適合宣言 (STC)



本ドキュメントは包括的なガイドではありません。以下の手順に従っても、デバイスのコンFORMANCEテストを完了できることを保証するものではありません。

このセクションの目的は、コンFORMANCEテストおよび適合宣言 (STC) とAnybusオブジェクトとの関係を説明することにあります。下記のオブジェクトは、ホストアプリケーションとAnybus CompactComには存在しますが、EtherNet/IPインターフェースには存在しません。オブジェクトに関する説明は、『Anybus CompactCom 40 Software Design Guide』 および『Anybus CompactCom 40 EtherNet/IP Network Guide』に記載されています。

テストを実施する前に、「CIPプロトコルテスト仕様」と「EtherNet/IPテスト仕様」を読むことを推奨します。これらのドキュメントには期待される応答や許容される動作様態が記載されており、多くの初期エラーを回避するのに役立ちます。早い段階で適合宣言とホストアプリケーションに修正を加えることができるため、時間と労力を削減することができます。

F.3.1 ホストオブジェクトの実装

最終製品がコンFORMANCEテストに確実に合格できるよう、ホストオブジェクトの実装を適応させる必要がある場合があります。CTソフトウェアを使用し、以下の手順に従ってください。ここでは、EtherNet/IPに関連するホストオブジェクトについてのみ説明します。

EtherNet/IPホストオブジェクト (F8h)

EtherNet/IPホストオブジェクト (F8h) の実装は、次のオブジェクトに影響します。アイデンティティオブジェクト (01h、CIPオブジェクト)、アセンブリオブジェクト (04h、CIPオブジェクト)、ポートオブジェクト (F4h、CIPオブジェクト)、CIPポートコンフィグレーションオブジェクト (ホストオブジェクト、0Dh)。また、STCの設定にも影響が及びます。以下に示すインスタンスアトリビュートについて考慮が必要です。

EtherNet/IPホストオブジェクト (F8h) - アトリビュート#1 - Vendor ID

ベンダーIDがCTソフトウェアならびにSTCのベンダー名と一致していなければなりません。

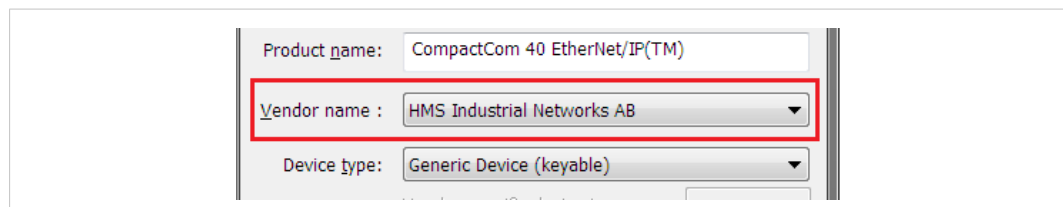



図 12

初めてテストをする場合は、テストソフトウェアが更新されていることが確かではないため、EtherNet/IPベンダーの名前がドロップダウンメニューの中にあることがあります。コンFORMANCEテストを実際に実施する前にデバイスとSTCの両方でベンダーIDを変更するのであれば、ベンダーID不一致によって報告されるエラー数を低減するため、リスト内の任意のベンダーIDを使用して事前テストをすることができます。

または、ベンダーはベンダー情報をVID.datファイルに追加することができます。

EtherNet/IPホストオブジェクト (F8h) - アトリビュート#2 - Device Type

Device Type はドロップダウンリストのDevice Typeと一致していなければなりません。

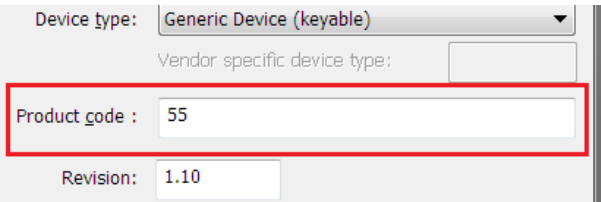


The screenshot shows a dropdown menu for 'Device type' with 'Generic Device (keyable)' selected. A red box highlights the dropdown menu. Below it is a text field for 'Vendor specific device type'.

図 13

EtherNet/IPホストオブジェクト (F8h) - アトリビュート#3 - Product Code

Product Code はドロップダウンリストのProduct Codeと一致していなければなりません。

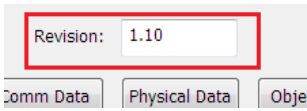


The screenshot shows a form with 'Device type' set to 'Generic Device (keyable)'. Below it, the 'Product code' field contains the value '55' and is highlighted with a red box. The 'Revision' field contains '1.10'.

図 14

EtherNet/IPホストオブジェクト (F8h) - アトリビュート#4 - Revision

RevisionはRevisionフィールドの<major>. <minor> と一致していなければなりません。



The screenshot shows a 'Revision' field with the value '1.10' highlighted by a red box. Below the field are buttons for 'Comm Data', 'Physical Data', and 'Obj'.

図 15

EtherNet/IPホストオブジェクト (F8h) - アトリビュート#5 - Serial Number

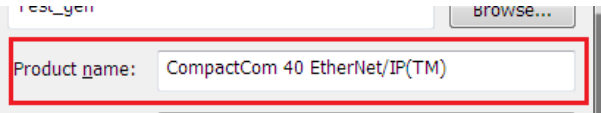
現行バージョンのCTテストでは、シリアル番号はチェックされません。



CIP仕様に従い、ベンダーIDとシリアル番号は一意である必要があります。HMSベンダーID (005Ah) と組合せてカスタムシリアル番号を使用することは許可されていません。

EtherNet/IPホストオブジェクト (F8h) - アトリビュート#6 - Product Name

Product NameはProduct Nameフィールドと一致していなければなりません。



The screenshot shows a 'Product name' field with the value 'CompactCom 40 EtherNet/IP(TM)' highlighted by a red box. A 'BROWSE...' button is visible to the right of the field.

図 16

EtherNet/IP ホストオブジェクト (F8h) - アトリビュート#7 - Producing Instance No.

Producingインスタンスはアセンブリオブジェクトに影響するため、リスト化する必要があります。ほとんどのアプリケーションで、Producingインスタンスは静的入力です。

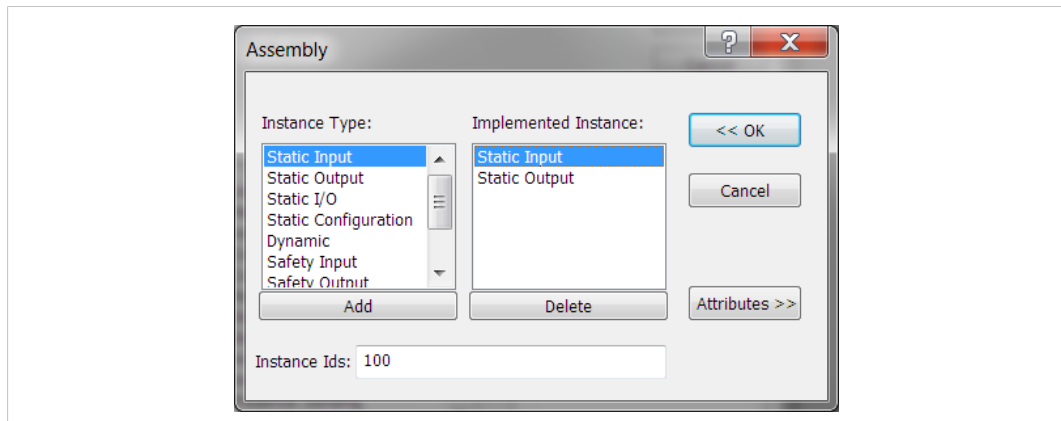


図 17

Producingインスタンスは、コネクシオンマネージャオブジェクトのコネクシオンにも影響します。正しいインスタンスを指向するよう、各コネクシオンのコネクシオンパスに修正を加える必要があります。下図は0x64をProducingインスタンスとしてリストしている例を示します。コネクシオンパスのエンコーディングについては、*Volume 1: Common Industrial Protocol Specification appendix C*を参照してください。

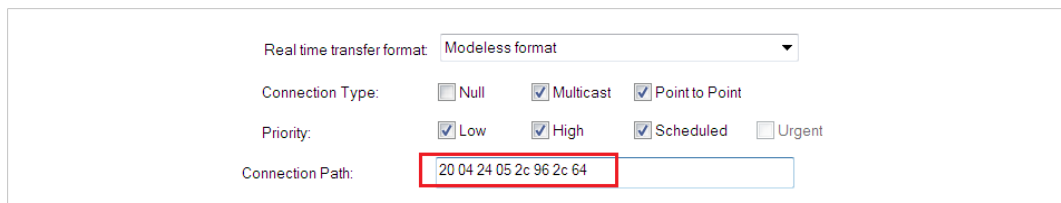


図 18



ホストアプリケーションがアセンブリマッピングホストオブジェクトを実装している場合、このアトリビュートは配列となります。このオブジェクトについて詳しくは、下記を参照してください。

EtherNet/IPホストオブジェクト (F8h) - アトリビュート#8 - Consuming Instance No.

この応答フィールドはアセンブリオブジェクトに影響します。ほとんどのアプリケーションで、Producingインスタンスは静的出力です。

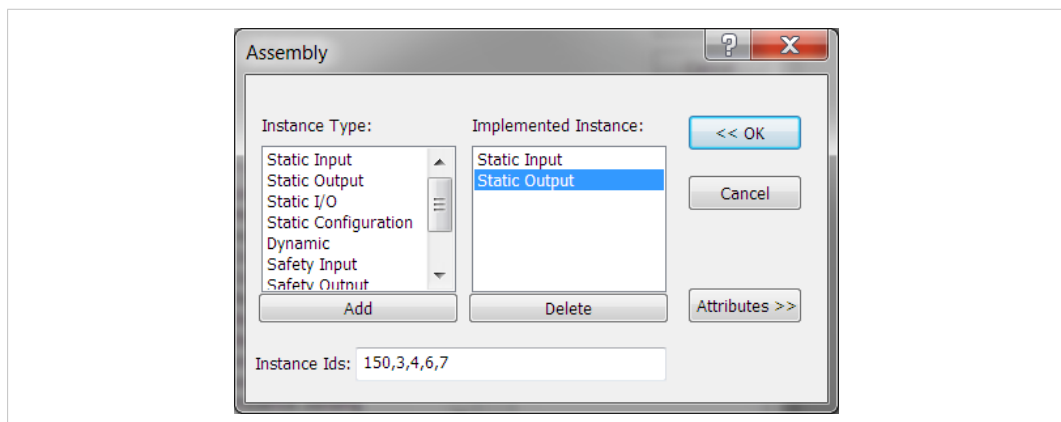


図 19

Producingインスタンスは、コネクシオンマネージャオブジェクトのコネクシオンにも影響します。正しいインスタンスを指向するよう、各コネクシオンのコネクシオンパスに修正を加える必要があります。

下図は0x96をConsumingインスタンスとしてリストしている例を示します。コネクションパスのエンコーディングについては、*Volume 1: Common Industrial Protocol Specification appendix C*を参照してください。

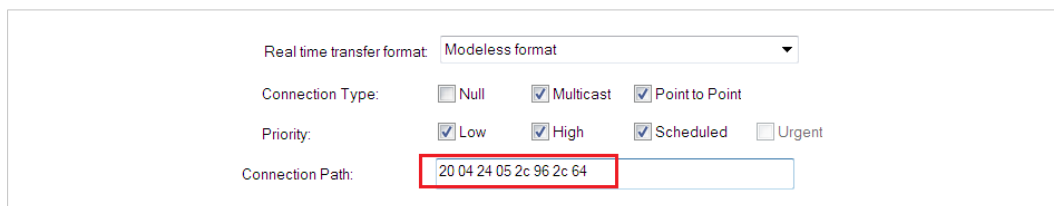


図 20

i ホストアプリケーションがアセンブリマッピングホストオブジェクトを実装している場合、このアトリビュートは配列となります。このオブジェクトについて詳しくは、下記を参照してください。

EtherNet/IPホストオブジェクト (F8h) - アトリビュート#9 - Enable Communication Settings from Net

このアトリビュートは、ネットワーク上のその他のデバイスに対して、CIP TCP/IPオブジェクトとCIPイーサネットリンク オブジェクトへのアクセスを使用した通信設定の調整ができるようにする設定をします。この方式がサポートされている場合は、物理データセクションのボックスを以下のようにオンにしてください。

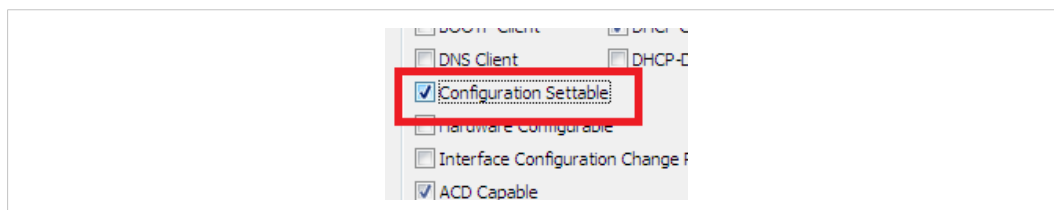


図 21

影響を受けるTCP/IPインターフェースオブジェクトおよびイーサネットリンク オブジェクトのアトリビュートの設定が可能であることをサポートするには、STCファイルを下表に従って設定する必要があります。

Enable communication settings from Net	CIP TCP/IPオブジェクト	CIPイーサネットリンクオブジェクト
True	アトリビュート3、5、6、8、9、10、11、12のサービスを有効に設定。	アトリビュート6は有効。 サービスをオブジェクトレベルで有効に設定。
False	アトリビュート3、8、9、10、11、12のサービスを有効に設定。 アトリビュート5、6のサービスを無効に設定。	アトリビュート6は無効。 サービスをオブジェクトレベルで無効に設定 (Admin stateアトリビュートが実装されている場合は不適用)。 ハードウェアスイッチを使用する場合を除き、アトリビュート3はすべてのモードで設定可能。

DLR機能を使用するには、イーサネットポートの強制設定が可能 (10 Mbps、半二重など) である必要がありますのでご注意ください。ネットワークからの通信設定が無効になっているDLRデバイスの場合、イーサネットの設定を強制的に行えるようにする機能がアプリケーションから提供されなければなりません。例として、ユーザーがイーサネットの設定を強制的に行えるようにする機能をWebサーバに持たせることができます。

EtherNet/IPホストオブジェクト (F8h) - アトリビュート#11 - Enable CIP Forwarding

CIP Forwardingを有効にすると、ホストアプリケーションは、CompactComによって実装されていないCIPオブジェクトとAnybus アセンブリオブジェクトのインスタンスとの両方に対する、すべての要求に回答します。Conformance Testソフトウェアは、要求がアプリケーションによって正しく処理されているか否かをチェックします。対応するEtherNet/IPネットワークの付録中のCIPオブジェクトに関するセクションにリストされているオブジェクトとアセンブリオブジェクトインスタンスは、すべてAnybus CompactComによって処理されます。これは、CIP Forwardingが有効になっている場合、CIPオブジェクトセクションにリストされていないCIPオブジェクトおよびインスタンスに対する要求は、すべてProcess_CIP_Requestコマンドに対する応答で処理する必要があることを意味します。

『CIP Network Libraries Volume 1』で定義されたデバイスプロファイルをユーザーがサポートしている場合、CIP Forwardingを有効にする必要があります。また、CIPネットワーク仕様で指定されていないオブ

ジェクトおよびアセンブリオブジェクトインスタンスについては、これらがベンダー固有の範囲内にある限り、ベンダーはオブジェクトとアセンブリオブジェクトインスタンスを定義およびサポートできます。

CIPで定義されたデバイスプロファイルは、どのオブジェクト・インスタンス・アトリビュートとアセンブリオブジェクトのインスタンスがデバイスによるサポートを必要とするのかを定義します。また、必須のサービスと動作様態が定義されます。デバイスプロファイルについては、『CIP Network Libraries Volume 1』の第6章に詳しい説明があります。Anybus CompactComでサポートされているデフォルトプロファイルは、ジェネリックデバイス(Keyable)です。このデバイスプロファイルならびにその他一部のプロファイルは追加のオブジェクトインスタンスやアセンブリインスタンスをサポートする必要がないため、CIP forwardingを有効にする必要はありません。

「適切な応答」は、どのような要求がどのようなときに行われたかによって、その意味が異なります。複雑さを軽減するための推奨事項を以下に示します。

- どのような組合せでオブジェクト、インスタンス、アトリビュートを実装するかを決定する。必須、オプション、ベンダー固有の組合せが正しいことを仕様を見て確認する。
- 実装した組合せに対して、どのサービスをサポートするかを決定する。必須、オプション、ベンダー固有のサービスが正しく実装されていることを仕様を見て確認する。
- 実装されたサービスとパスとの正しい相互作用のために、適切なアプリケーションの動作が提供されていることを確認する。
- アプリケーションでサポートされていないすべてのパスに対して、正しいエラー応答が提供される。

アプリケーションがAnybus CompactComエラーコード「Unsupported Object (3)」または「Unsupported Instance (4)」のいずれかを返すと、CIPステータスコード0x05 (宛先不明のパス) が応答される。ステータスコードについては『CIP Network Libraries Volume 1』の付録Bを、CIPエラーコードからAnybus CompactComエラーコードへの変換については『Anybus CompactCom EtherNet/IP Network Guide』のCIPオブジェクトに関するセクションを参照してください。

- サポートされていないコマンドすべてに対して、正しいエラー応答が提供される。

アプリケーションがAnybus CompactComエラーコード「Unsupported Command (5)」を返すと、CIPステータスコード0x08 (サービスがサポートされていない) が応答されます。ステータスコードについては『CIP Network Libraries Volume 1』の付録Bを、CIPエラーコードからAnybus CompactComエラーコードへの変換については『Anybus CompactCom EtherNet/IP Network Guide』のCIPオブジェクトに関する章を参照してください。

- 変数を変更するすべてのコマンドに対してエラーチェックを行い、CIPによって定義された正しいエラーコードで応答する。ステータスコードについては『CIP Network Libraries Volume 1』の付録Bを、CIPエラーコードからAnybus CompactComエラーコードへの変換については『Anybus CompactCom EtherNet/IP Network Guide』のCIPオブジェクトに関する章を参照してください。

プロファイルの要件、サービス、動作様態のほか、CIP Network Librariesで指定されているオブジェクト定義を確認してください。

EtherNet/IPホストオブジェクト (F8h) - アトリビュート#12 - Enable Parameter Object

CIPパラメータオブジェクトの目的は、デバイスコンフィグレーションのために統一されたインターフェースを提供することです。EtherNet/IPでは、設定可能なパラメータごとにパラメータオブジェクトのインスタンスが1つ必要です。CIPパラメータオブジェクトへの要求は、ホストアプリケーションデータオブジェクトへの要求に変換されます。この要求にFALSEと応答することで、パラメータオブジェクトへのアクセスを無効にすることができます。必須またはオプションのインスタンスアトリビュートについては、『CIP Network Libraries Volume 1』の表5A-14.7を参照してください。オブジェクトが無効になっている場合、STCファイル内のサポートされているオブジェクトリストからそのパラメータオブジェクトを削除する必要があります。

EtherNet/IPホストオブジェクト (F8h) - アトリビュート#13 - Input-Only Heartbeat Instance Number

デフォルトでは、このインスタンス番号は3です。この値をデフォルトから変更するには、.stcファイルのコネクションマネージャの部分にリストされているInput onlyコネクションを変更する必要があります。下図は、Input-onlyコネクションについて、03がコネクションパスの0->Tコネクションポイントとして表されている状態を示します。コンフィグレーションパスに関する説明は、『CIP Networks Library Volume 1』の付録Cを参照してください。

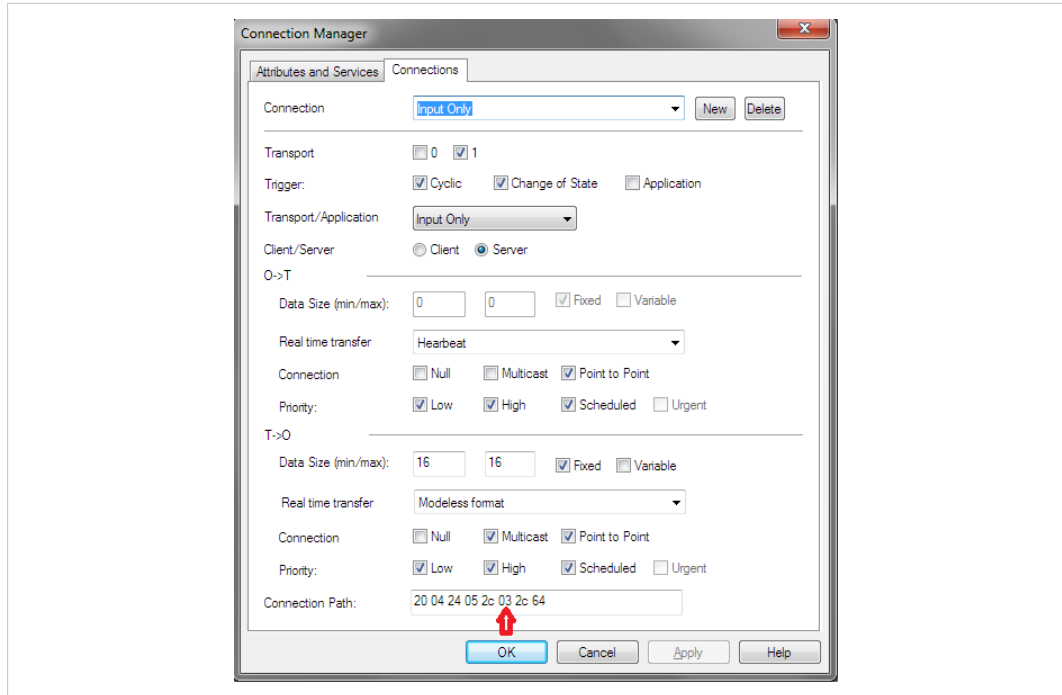


図 22

EtherNet/IPホストオブジェクト (F8h) - アトリビュート#14 - Listen-Only Heartbeat Instance Number

このアトリビュートは、(オリジネータからターゲットへの) ハートビート接続ポイントのアセンブリインスタンスを設定します。このインスタンスは、Input-only接続のコネクションマネージャオブジェクトに接続ポイントとしてリストされている必要があります。デフォルトのインスタンス番号は4です。

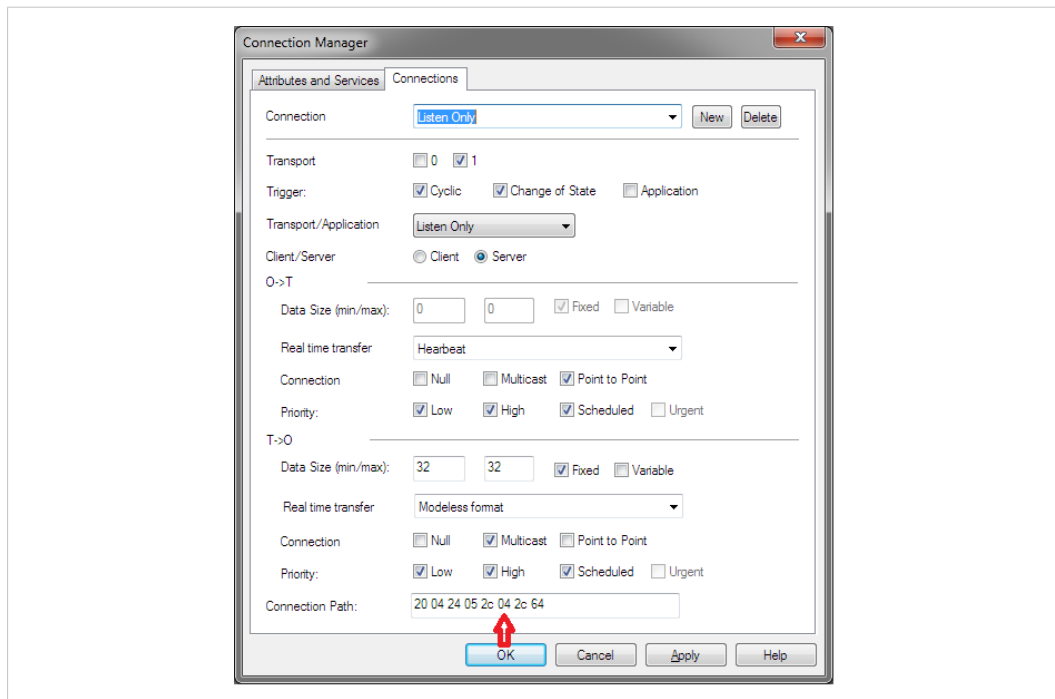


図 23

EtherNet/IPホストオブジェクト(F8h) - アトリビュート15 - Assembly Object Configuration Instance Number

デバイスのコンフィグレーションパラメータは、アセンブリインスタンス内でグループ化できます。デフォルトでは、このインスタンスは5です。コンフィグレーションインスタンスのサポートは、EtherNet/IPホストオブジェクトのGet_Configuration_InstanceおよびSet_Configuration_Instance機能によって提供されます。このインスタンスを使用してコンフィグレーションデータが渡される場合、このアセンブリを静的コンフィグレーションとしてアセンブリオブジェクトにリストする必要と、コネクションマネージャのコネクションパスにリストする必要があります。

EtherNet/IPホストオブジェクト (F8h) - アトリビュート16 - Disable Strict IO Match

デバイスのコンフィグレーションパラメータは、アセンブリインスタンス内でグループ化できます。デフォルトでは、このインスタンスは5です。コンフィグレーションインスタンスのサポートは、EtherNet/IPホストオブジェクトのGet_Configuration_InstanceおよびSet_Configuration_Instance機能によって提供されます。このインスタンスを使用してコンフィグレーションデータが渡される場合、このアセンブリを静的コンフィグレーションとしてアセンブリオブジェクトにリストする必要と、コネクシオンマネージャのコネクシオンパスにリストする必要があります。

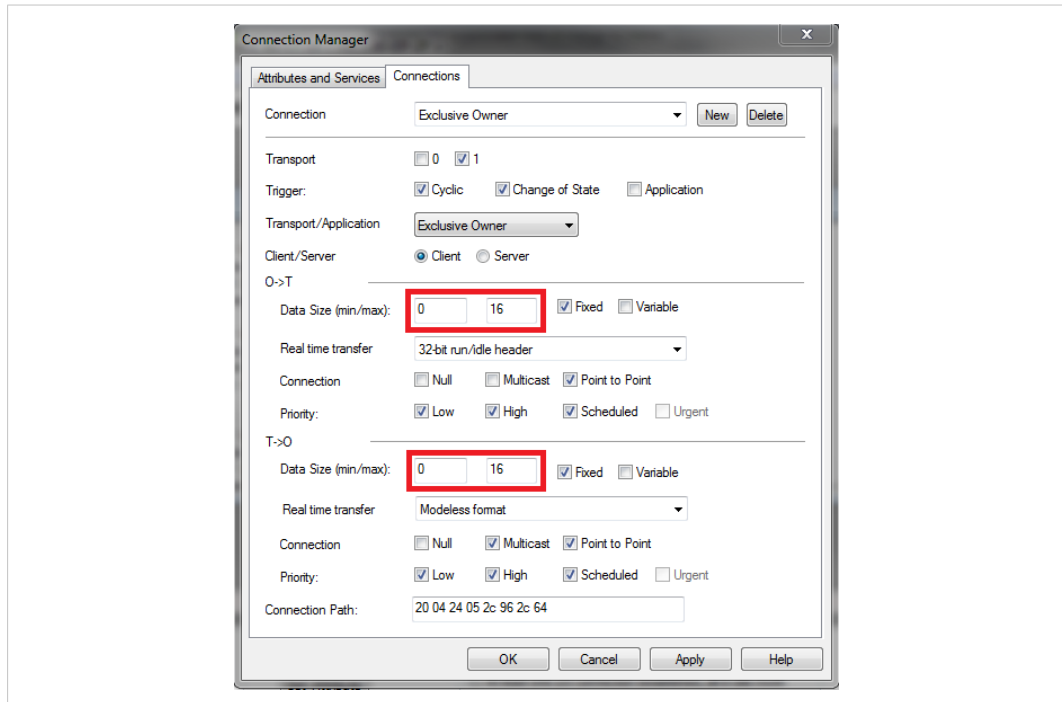


図 24

EtherNet/IPホストオブジェクト (F8h) - アトリビュート17 - Enable Unconnected Routing

このアトリビュートを有効にすると未接続のルーティングが可能になり、CIPポートオブジェクト (F4h) へのアクセスが許可されます。オリジネータはCIPルーティングを使用し、Anybus CompactComを介して他のサブネットやバックプレーンにリンクすることが可能になります。EtherNet/IPの場合、複数のポートオブジェクトインスタンスがシングルまたはデュアル物理ポートを共有できます。CIPルーティングが可能な各ポートに対して、CIPポートオブジェクトのインスタンスが1つ存在する必要があります。このアトリビュートを有効にするには、アプリケーションがホストのCIPポートコンフィギュレーションオブジェクト (ODh) をサポートしている必要もあります。通信データセクションのチェックボックスを選択すると、ポートオブジェクトが実装済みオブジェクトリストに追加され、適合宣言ファイルをポート転送機能が反映された設定にすることができます。

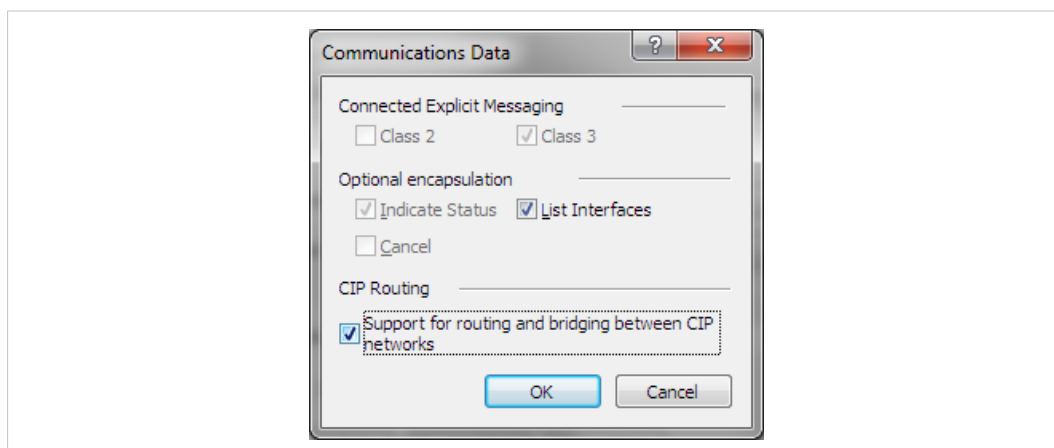


図 25

EtherNet/IPホストオブジェクト (F8h) - アトリビュート18 - Input-Only Extended Heartbeat Instance Number

Input-onlyハートビート接続の拡張バージョンは、1つの例外を除いて機能的に同じです。接続の状態はモジュールの状態に影響を与えません。すなわち、この接続のタイムアウトはモジュールを強制的にExceptionの状態にしません。このインスタンスはコネクションポイントに使用できます。コネクションマネージャで当該のコネクションにラベルを付けておく必要があります。インスタンス番号がコネクションパス内に表されている必要があります。

EtherNet/IPホストオブジェクト (F8h) - アトリビュート19 - Listen-Only Extended Heartbeat Instance Number

Listen-onlyハートビート接続の拡張バージョンは、1つの例外を除いて機能的に同じです。接続の状態はモジュールの状態に影響を与えません。すなわち、この接続のタイムアウトはモジュールを強制的にExceptionの状態にしません。このインスタンスはコネクションポイントに使用できます。コネクションマネージャで当該のコネクションにラベルを付けておく必要があります。インスタンス番号がコネクションパス内に表されている必要があります。

EtherNet/IPホストオブジェクト (F8h) - アトリビュート20 - Interface label port 1

このラベルはCTテストソフトウェアではチェックされませんが、変更した場合はEDSファイルが同等の文字列で更新されていることを確認してください。

EtherNet/IPホストオブジェクト (F8h) - アトリビュート21 - Interface label port 2

このラベルはCTテストソフトウェアではチェックされませんが、変更した場合はEDSファイルが同等の文字列で更新されていることを確認してください。

EtherNet/IPホストオブジェクト (F8h) - アトリビュート22 - Interface label internal port

このラベルはCTテストソフトウェアではチェックされませんが、変更した場合はEDSファイルが同等の文字列で更新されていることを確認してください。

EtherNet/IPホストオブジェクト (F8h) - アトリビュート26 - Enable EtherNet/IP QuickConnect

QuickConnectを有効にすると、電源投入からネットワーク上で使用可能になるまでの時間が可能な限り短くなります。QuickConnectはデフォルトのEDSファイルに変更を一つ加える必要があります。また、2ポートモジュールの場合、ポート1とポート2にデバイス上で外部ラベルが付与されている必要もあります。

QuickConnectが有効になっている場合、TCP/IPインターフェースオブジェクトのアトリビュート#12をSTCファイルでSetおよびGetアクセスに設定する必要があります。QuickConnectが無効になっている場合は、SetおよびGetアクセスにチェックマークが付いてはなりません。

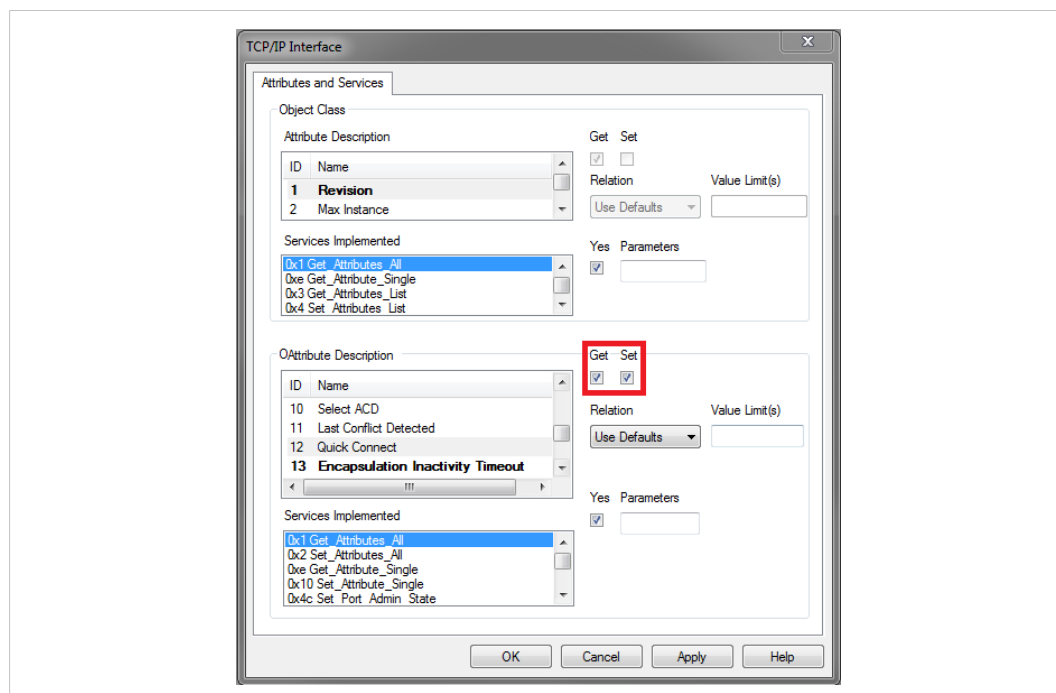


図 26

EtherNet/IPホストオブジェクト (F8h) - アトリビュート29 - Ignore Sequence Count Check

この機能を有効にすると、CIP Network Librariesの仕様に違反します。

EtherNet/IPホストオブジェクト (F8h) - アトリビュート30 - ABCC ADI Object Number

Anybus CompactCom EtherNet/IP製品群のモジュールは、デフォルト動作様態としてCIP ADIオブジェクト (A2h) を介したアプリケーションデータオブジェクトのインスタンスへのアクセスを提供します。このオブジェクトクラス番号を変更したり、アクセスを完全に無効にしたりすることができます。A2hはベンダー固有の範囲内にあるため、ベンダーが独自のオブジェクトを自由に実装できることにご注意ください。ベンダー固有の範囲外にあるオブジェクトクラス番号の選択は、デバイスがそのオブジェクトによって指定された機能を提供し、CIP Networks Libraryで設定されているアトリビュートとサービスの編成に忠実に従っている場合にのみ行うようにしてください。ベンダー固有の範囲は64h - C7hと300h - 4FFhです。

EtherNet/IPホストオブジェクト (F8h) - アトリビュート31 - Enable DLR

Anybus CompactCom EtherNet/IPのデフォルト動作様態では、DLRは有効になっています。何らかの理由でDLRが無効になっている場合、そのDLRオブジェクトをSTCファイル内のサポートされているオブジェクトリストから削除する必要があります。

Ethernet ホストオブジェクト (F9h)

このオブジェクトの属性の多くはODVAの仕様に含まれず、コンフォーマンステストにまったく関係がないため、このドキュメントには記載されていません。

Ethernetホストオブジェクト (F9h) - 属性#1 - MAC Address (属性#8および#9も同様)

MACアドレスが適合宣言に記載されている必要があります。この属性には物理データセクションでアクセスできます。

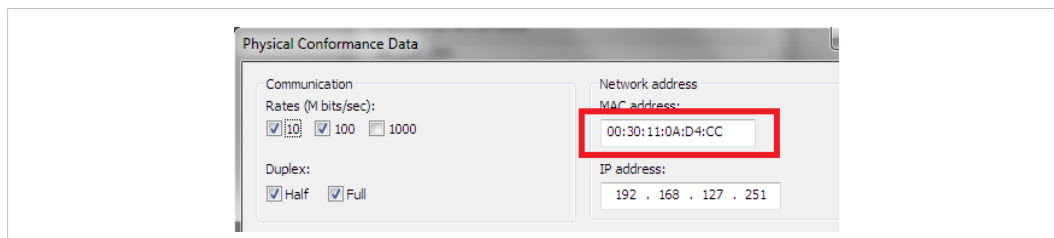


図 27

Ethernetホストオブジェクト(F9h) - 属性#11 - Enable ACD (ACD = 自動衝突検出)

MACアドレスが適合宣言に記載されている必要があります。この属性には物理データセクションでアクセスできます。

Auto Collision Detectionは、複数のデバイスが同じIPアドレスを持つことによるエラーを検出・軽減するEtherNet/IPの機能です。この属性には物理データセクションでアクセスできます。また、ODVAのコンフォーマンステスト詳細フォームには、デバイスがACD対応かどうかを示すセクションがあります。

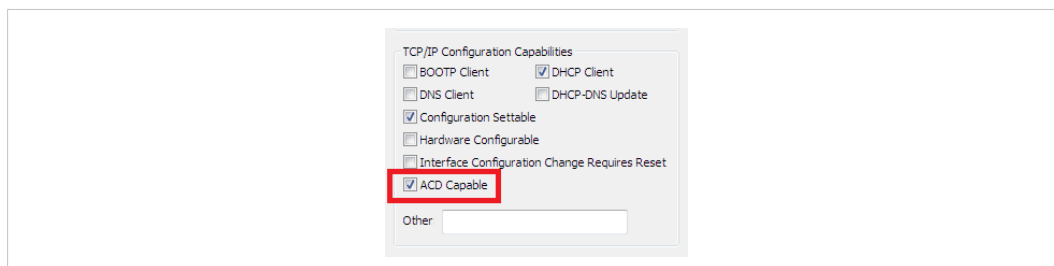


図 28

Ethernetホストオブジェクト(F9h) - 属性13 - Port 2 State

Anybus CompactCom B40 およびC40アプリケーションでは、アプリケーションにイーサネットポート1つ分のスペースしかない場合や、何らかの理由でイーサネットポートが1つしか必要でない場合、ポート2が搭載されない場合があります。この属性が非アクティブに設定されているとDLRオブジェクトは自動的に無効になるため、STCファイル内のサポートされているオブジェクトリストからこのオブジェクトを削除する必要があります。

Ethernetホストオブジェクト(F9h) - アトリビュート#24 - Enable DHCP Client

何らかの理由でホストアプリケーションにDHCPをサポートさせたくない場合、このアトリビュートはFalseに設定します。STCファイルの物理データセクションにあるDHCPクライアントサポートのチェックマークを外す必要があります。

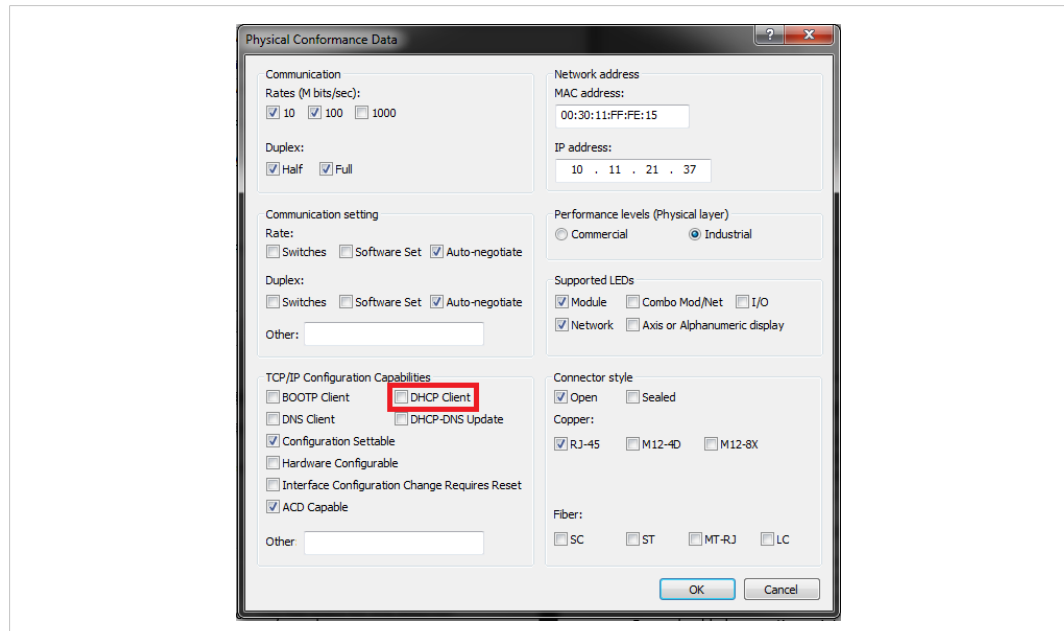


図 29

CIP Identity ホストオブジェクト (EDh)

このオブジェクトは、デフォルトでサポートされている第1インスタンスを超えて、アイデンティティオブジェクト (CIPオブジェクト、01h) の追加インスタンスをデバイスにサポートさせることができます。アイデンティティオブジェクトの追加インスタンスに対するサポートは、Identity Classアトリビュートの最大インスタンスとインスタンス数を適切な値に変更して、適合宣言で反映されるようにする必要があります。

アセンブリマッピングオブジェクト (EBh)

アセンブリマッピングホストオブジェクトを使用すると、最大で6つのProducing CIPアセンブリインスタンス、6つのConsuming CIPアセンブリインスタンスを作成することができます。これらの追加アセンブリは、Anybus CompactCom40 EtherNet/IPにおける標準的なExclusive Ownerコネクションと同等のコネクションポイントをコネクションマネージャ内にも作成します。

追加されたコネクションポイントについては、コネクションポイントごとにSTCファイル内のアセンブリオブジェクトとコネクションマネージャオブジェクトを更新して、これらの新しいアセンブリインスタンスを記述する必要があります。

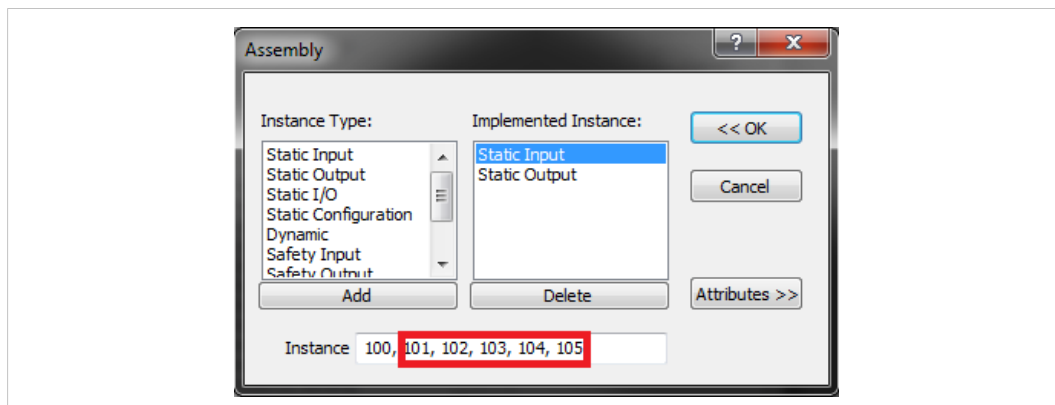


図 30

この例では、Producingアセンブリが5つ追加されている例を示します。

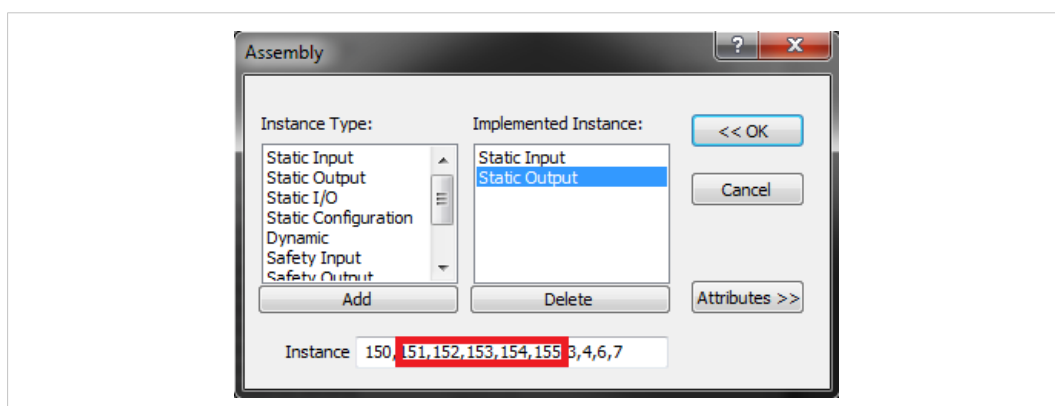


図 31

この例では、Consumingアセンブリが5つ追加されている例を示します。

STCファイルのコネクションマネージャのセクションで、ProducingコネクションポイントとConsumingコネクションポイントとの間で可能なコネクションの組合せをすべてリストする必要はありません。どのコネクションの組合せを顧客が使用できるかは、ベンダーがEDSファイル内で決めることができますが、EDSファイルのコネクションマネージャのセクションに、EDSファイル内に存在するすべてのコネクションをリストする必要があります。

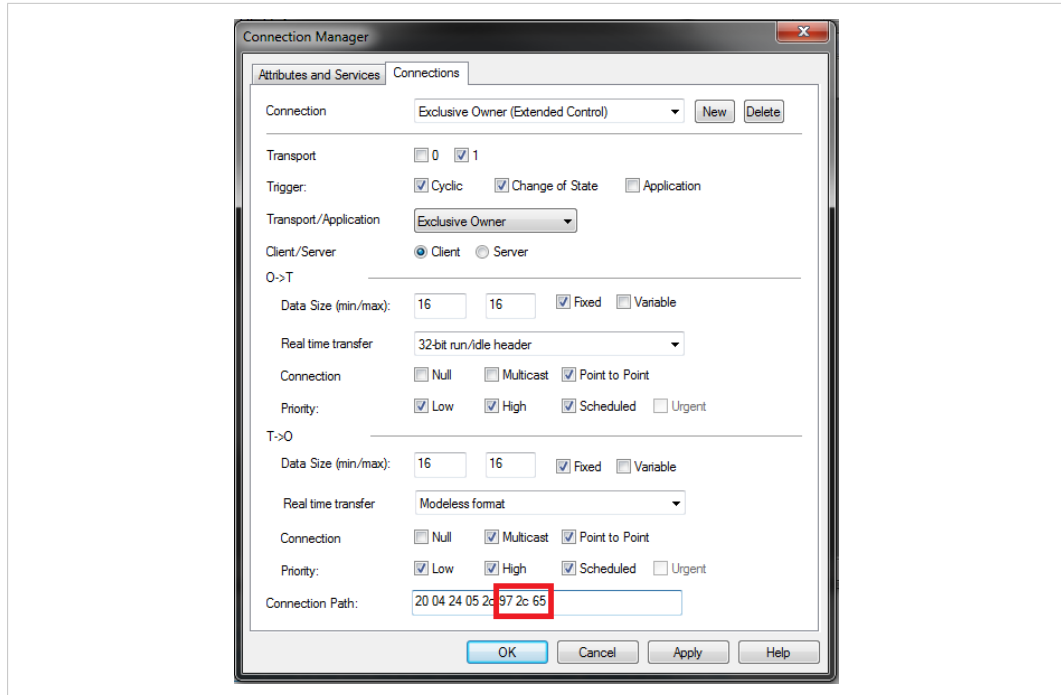


図 32

上図は、追加されたExclusive Ownerコネクションがコネクションポイント101/151に接続している例を示します。

i コンフォーマンステストに合格するには、アセンブリマッピングホストオブジェクトのWrite_Assembly_DataおよびRead_Assembly_Data サービスをアプリケーションに実装する必要があります。

F.3.2 Anybusモジュールオブジェクトの実装

ここでは、EtherNet/IPに関連するAnybusモジュールオブジェクトについてのみ説明します。

Network オブジェクト (03h)

ネットワーク オブジェクト (03h) - アトリビュート#5 - Write Process Data Size

Write Process Data SizeはT->Oコネクションにマッピングされた現在のデータ量を表します。必ずとは限りませんが、ほとんどの場合でこの値はコネクションマネージャのT->Oコネクションサイズと一致します。プロセスデータのリマッピングをサポートすることで、コネクションポイントに対して複数のアセンブリのサポートが可能になります。

ネットワークオブジェクト (03h) - アトリビュート#6 - Read Process Data Size

Read Process Data SizeはO->Tコネクションにマッピングされた現在のデータ量を表します。必ずとは限りませんが、ほとんどの場合でこの値はコネクションマネージャのO->Tコネクションサイズと一致します。プロセスデータのリマッピングをサポートすることで、コネクションポイントに対して複数のアセンブリのサポートが可能になります。

ネットワークオブジェクト (03h) - CIPポートコンフィグレーションオブジェクト (0Dh)

CIPルーティングは、EtherNet/IPホストオブジェクトのインスタンスアトリビュート#17によって有効にできます。このオブジェクトの各インスタンスはCIPポートオブジェクト (F4h) のインスタンスに対応します。

ただし、CIPポートが必ずしも物理ポートに対応するとは限りません。2ポートのAnybus CompactComの場合、2つのネットワークコネクタが1つのCIPルーティング可能ポートに対応します。CIPポートが1つのデバイスの場合はポートオブジェクトのサポートは不要ですが、通信アダプタデバイスのプロファイルについてはこのオブジェクトのサポートが必要となります。

G 後方互換性

産業用ネットワークモジュールのAnybus CompactCom 40シリーズは、Anybus CompactCom 30シリーズよりも性能が大きく向上しており、機能も追加されています。40シリーズは30シリーズに対する後方互換性があり、30シリーズ用に開発されたアプリケーションは、大幅な変更を施さずに40シリーズでも使用することが可能です。また、同じアプリケーションで30シリーズと40シリーズのモジュールを混在させることも可能です。

ここでは、1つのアプリケーションを両シリーズに適用できるように設計する場合や、30シリーズ用のアプリケーションを40シリーズ用に適応させる場合にAnybus CompactCom 40 EtherNet/IPについて考慮すべき後方互換性の問題について説明します。

G.1 初期段階における注意事項

Anybus CompactCom 30シリーズのモジュール用に開発されたホストアプリケーションを、40シリーズのモジュールとも互換性を持つように変更する作業を開始する際には、次の2点を考慮する必要があります。

- できる限り少ない作業で、すなわち可能な限り現在の設計を再利用して、実装を追加します。
 - これが最も迅速で簡単なソリューションですが、40シリーズで利用可能な多くの新機能 (高速通信インターフェース、拡大されたメモリ領域、高速通信プロトコルなど) を有効化できないという欠点があります。
 - ホストアプリケーションが40シリーズのモジュールと互換性があることを確認するには、以下のハードウェアとソフトウェアの相違を確認する必要があります。現在の設計に小規模な変更が必要になる場合があります。
- 再設計を行って、40シリーズに用意されている新機能すべてが利用できるようにします。
 - 新しい通信プロトコルをサポートする新しいドライバとホストアプリケーションのサンプルコードは、www.anybus.com/starterkit40で入手できます。このドライバは、30シリーズおよび40シリーズ両方のモジュールをサポートします。
 - ホストアプリケーションが40シリーズのモジュールと互換性があることを確認するには、以下のハードウェアの相違を確認する必要があります。



この情報は、30シリーズと40シリーズとの相違のみを対象としています。

サポートページへのリンク: www.anybus.com/support

G.2 ハードウェアの互換性

Anybus CompactComは3つのハードウェア形態、モジュール、チップ、およびブリックで利用できます。

G.2.1 モジュール

30シリーズおよび40シリーズのモジュールは、寸法、外形、コネクタ、LEDインジケータ、取付部品などの物理的特性を共有しています。また、両シリーズ共に、ハウジングなしモジュールも使用可能です。



図 33 Anybus CompactCom M30/M40

G.2.2 チップ

Anybus CompactComのチップ (C30/C40) は、バージョン間で物理的な寸法が完全に異なります。



ハードウェアを大幅にアップデートしない限り、チップソリューションを30シリーズから40シリーズへ移行する方法はありません。

G.2.3 ブリック

Anybus CompactCom B40-1は、Anybus CompactCom B30と寸法を共有していません。そのため、B40-1は移行に適していません。しかし、HMS Industrial Networksは、移行に使用可能な40シリーズ用ブリックバージョンを別途開発しました。同製品B40-2は、B30と寸法などを共有しています。Anybus CompactCom B40-2の詳細についてはHMS Industrial Networksまでお問い合わせください。

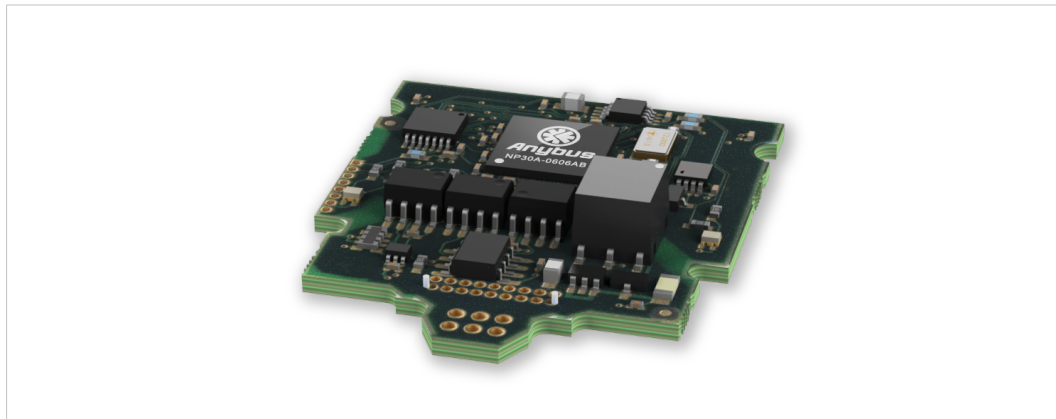


図 34 Anybus CompactCom B30

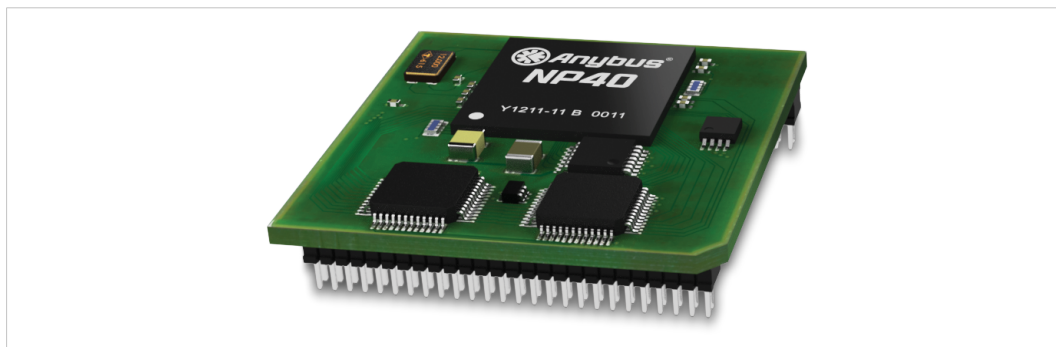


図 35 Anybus CompactCom B40-1 (移行用ではありません)

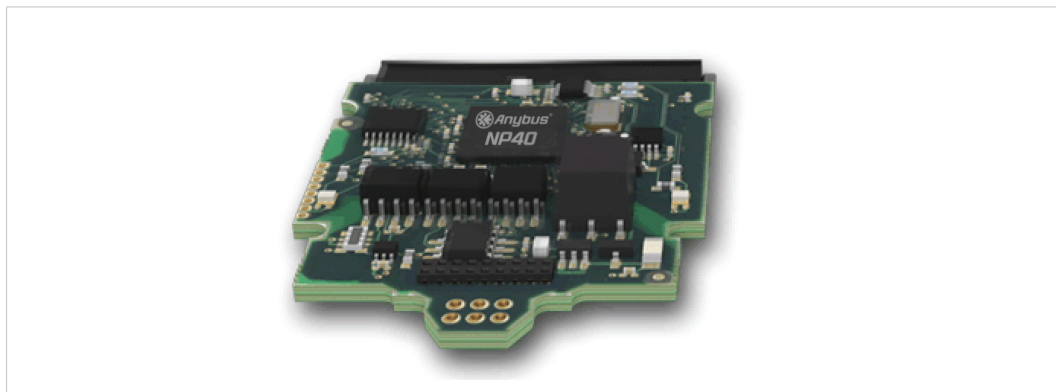


図 36 Anybus CompactCom B40-2

G.2.4 ホストアプリケーションインターフェース

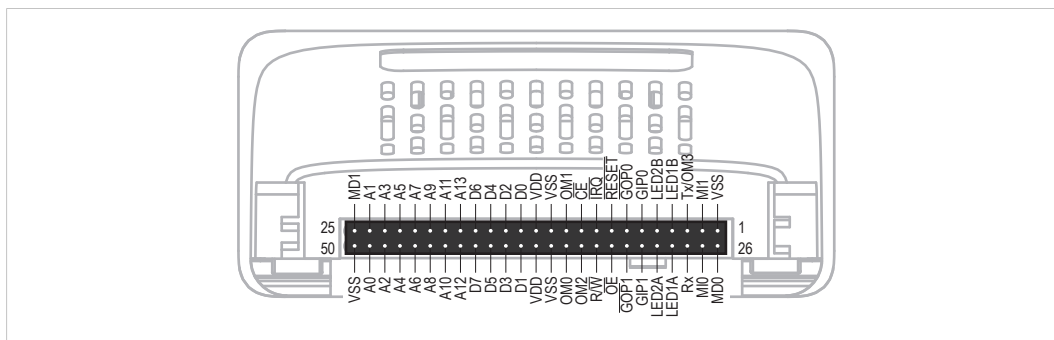


図 37

ホストアプリケーションインターフェースの信号の一部は機能性および/または機能が変更になっているため、これらの互換性をチェックする必要があります。以下のセクションを参照してください。

Tx/OM3

30シリーズでは、このピンはTxのためだけに使用されています。起動中は3ステートであり、初期化後にAnybus CompactCom UARTによって駆動されます。40シリーズでは、このピンは第4の動作モード設定ピン(OM3)として使用されます。リセット解除後の起動時にこのピンが読み取られて、使用する動作モードが決定されます。その後、ピンはTx出力に変更されます。

40シリーズでは、弱いプルアップ抵抗がこのピンに内蔵されています。30シリーズのモジュールまたはブリック上でこのピンが接続されていなかったり、Highにプルアップされていたり、ホストプロセッサのHigh-Zデジタル入力に接続されている場合、40シリーズとの互換性を保ちます。外付けのプルアップ抵抗は推奨されますが、必須ではありません。



このピンが30シリーズアプリケーションの起動時にホストによってLowレベルに引き下げられると、アプリケーションとして置き換えられた40シリーズのモジュールまたはブリックは、想定した動作モードに入りません。

関連情報: Anybus CompactCom M40 Hardware Design Guide (HMSI-216-126)、「アプリケーションコネクタピン概要」

モジュール識別 (MI[0..1])

これらのピンは、どの種類のAnybus CompactComが搭載されているかを特定するために、ホストアプリケーション(お使いの製品)によって使用されます。40シリーズと30シリーズでは、この識別信号が異なります。



ソフトウェアでこの識別信号を使用する場合は、新しい識別値を処理する必要があります。

MI1	MI0	モジュールの種類
LOW	LOW	Anybus CompactCom 30がアクティブ
HIGH	LOW	Anybus CompactCom 40がアクティブ

アプリケーションによるMI[0~1]のサンプリングは、起動からSETUP状態終了までの間のみです。起動時およびリセット解除前のピンはLowです。

関連情報: Anybus CompactCom M40 Hardware Design Guide (HMSI-216-126)、「Setting/Sync」

GIP[0..1]/LED3[A..B]

これらのピンは、30シリーズではデフォルトで3ステート入力となっています。40シリーズでは、NW_INIT状態まで3ステートです。その後はオープンドレイン方式のActive Low LED出力(LED3A/LED3B)となります。

現在の設計が以下の条件を満たしている場合、ハードウェアの変更は必要ありません。

- これらのピンがGNDに接続されている

- ピンのプルアップが行われている
- ピンのプルダウンが行われている
- ピンが未接続のままである

ただし、アプリケーションがピンをHighにすると、短絡が発生します。

ピンをLEDに接続する場合はプルアップ抵抗が必要です。

40シリーズでは、Anybusオブジェクト (01h) のアトリビュート#16 (GPIO構成) を使用して、GIP[0~1]とGOP [0~1]をハイインピーダンスの状態 (3ステート) に設定することが可能です。つまり、ホストアプリケーションのハードウェアを変更できない場合、このアトリビュートを使用して、NW_INIT状態を離れる前にGIPとGOPがハイインピーダンス状態になるように設定できます。

関連情報: *Anybus CompactCom M40 Hardware Design Guide (HMSI-216-126)*、 「LEDインターフェース/D8-D15 (データバス)」

GOP[0..1]/LED4[A..B]

これらのピンは、30シリーズではデフォルトにより出力 (High状態) になっています。40シリーズではNW_INIT状態まで3ステートで、その後、プッシュプル方式のActive Low LED出力 (LED4A/LED4B) となります。

この変化はお使いの製品に影響しません。

関連情報: *Anybus CompactCom M40 Hardware Design Guide (HMSI-216-126)*、セクション3.2.3、 「LEDインターフェース/D8-D15 (データバス)」

アドレスピンA[11..13]

アドレスピン11、12、13は30シリーズでは無視されます。後方互換性のある8ビットパラレルモードで40シリーズモジュールにアクセスする場合、これらのピンはHighでなければなりません。これらのピンが未接続になっている場合やGNDに接続されている場合は、ハードウェアの変更を行って、これらをHighにする必要があります。

最大入力信号レベル (V_{IH})

30シリーズの最大入力信号レベルは $V_{IH}=V_{DD}+0.2$ V、40シリーズでは $V_{IH}=3.45$ Vと指定されています。ロジックHighレベルで3.45Vを超えないようにしてください。

RMII互換性

RMIIモードがAnybus CompactCom 40モジュールで使用されており、30シリーズとの互換性維持を希望する場合、ピンにコンフリクトが生じるためAnybus CompactCom 30モジュールへの切り替え時に接続を無効にすることが重要です。ホストプロセッサのRMIIポートは、デフォルトで3ステートに設定し、RMII対応のAnybus CompactCom 40が検出された場合にのみ有効にする必要があります。ホストプロセッサの内部ハードウェア制御を通じてRMII接続を無効にできない場合、回路の短絡を防ぐために外部ハードウェア (FET/バススイッチ) を実装する必要があります。

関連情報: *Anybus CompactCom M40 Hardware Design Guide (HMSI-216-126)*、セクション3.2.5、「RMII – Reduced Media-Independent Interface」

G.3 ソフトウェア全般

G.3.1 拡張されたメモリ領域

40シリーズではメモリ領域が拡張されており、より大きいサイズのプロセスデータ (以前の最大256バイトに代わり最大4096バイト) およびメッセージデータ (以前の最大255バイトに代わり最大1524バイト) にアクセスできるようになりました。30シリーズには、アプリケーションでは使用できない、予約されたメモリ領域があります。40シリーズは、これらのメモリ領域の一部に新機能を実装しています。



拡張されたメモリ領域を使用するには、本章には記載のない新しい操作手順を実装する必要があります。

メモリ領域が特定のネットワークでサポートされていない場合、そのメモリ領域は使用できません。メモリの読み取り/書き込みテストなどの目的で、これらの領域にアクセスしないようにしてください。

関連情報: *Anybus CompactCom 40 Software Design Guide (HMSI-216-125)*、「メモリマップ」

G.3.2 より高速なピンポンプロトコル

40シリーズでは、ピンポンプロトコル (30シリーズで使用されているプロトコル) が高速化されています。30シリーズのモジュールは、通常10~100 μs内に、いわゆるping (ピン) に応答します。40シリーズは、通常2 μs内にpingに応答します。

割り込み駆動型のアプリケーション (パラレル動作モード) では、速度向上によりCPU負荷が増大する可能性があります。

G.3.3 起動時のAnybus CompactComからホストアプリケーションへの要求

ホストアプリケーション内のソフトウェアオブジェクトに対する要求はすべて、(オブジェクトが存在しない場合でも) 処理と応答が行われる必要があります。これは、30シリーズと40シリーズの両方に適用されます。40シリーズには、新機能のための追加オブジェクトが導入されています。

また、40シリーズによって既存のオブジェクトにコマンドが追加された場合も、(たとえサポートされていなくても) 応答が必要です。

処理不可能であってもすべてのコマンドに応答するという、しかるべき動作をする実装をお使いの場合は、何も変更する必要はありません。

G.3.4 Anybusオブジェクト (01h)

アトリビュート	30シリーズ	40シリーズ	変更/アクション/コメント
#1, Module Type	0401h	0403h	ホストアプリケーションが、40シリーズの新しいモジュールタイプの値を受け入れることを確認してください。
#15, Auxiliary Bit	利用可能	削除されました	40シリーズでは「Changed Data Indication」を無効にできません。下記の「コントロールレジスタCTRL_AUXビット」および「ステータスレジスタSTAT_AUXビット」も参照してください。
#16, GPIO Configuration	デフォルト: 一般的な入出力ピン	デフォルト: LED3およびLED4出力	以下も参照してください。 <ul style="list-style-type: none"> GIP[0..1]/LED3[A..B], ページ 218 GOP[0..1]/LED4[A..B], ページ 219

G.3.5 コントロールレジスタCTRL_AUXビット

- 30シリーズ 現在のテレグラムのプロセスデータが以前のものと比べて変化していることをコントロールレジスタのCTRL_AUXビットによってAnybus CompactComが示します。
- 40シリーズ CTRL_AUX ビットの値は常に無視されます。プロセスデータは常に受け入れられます。

Anybus CompactCom用にリリースされたAnybus CompactCom 30用サンプルドライバ全てに、この違いが当てはまります。

関連情報: *Anybus CompactCom 40 Software Design Guide (HMSI-216-125)*、「コントロールレジスタ」

G.3.6 ステータスレジスタSTAT_AUXビット

- 30シリーズ 現在のテレグラムの出力プロセスデータが以前のものと比べて変化していることをステータスレジスタのSTAT_AUXビットによって示します。この機能は、Anybusオブジェクト(01h)、アトリビュート#15で有効にする必要があります。デフォルトでは、STAT_AUXビット機能は無効になっています。
- 40シリーズ 以前のテレグラムに対して、ネットワークから送られた出力プロセスデータが更新されていることを(必ずしもデータが変更されている必要はありません)STAT_AUXビットが示します。この機能は常に有効になっています。

HMS Industrial NetworksからリリースされたAnybus CompactCom 30用サンプルドライバ全てに、この違いが当てはまります。

関連情報: *Anybus CompactCom 40 Software Design Guide (HMSI-216-125)*、「ステータスレジスタ」

G.3.7 コントロールレジスタCTRL_Rビット

- 30シリーズ このビットは、アプリケーションによっていつでも変更される可能性があります。
- 40シリーズ 8ビットパラレル動作モードでは、STAT_Mビットがステータスレジスタにセットされる時、このビットは1から0への遷移しか許可されません。シリアル動作モードを使用している場合は、最終を示す空フラグメント直後のテレグラム内で、1から0への遷移も許可されます。

HMS Industrial NetworksからリリースされたAnybus CompactCom 30用サンプルドライバ全てに、この違いが当てはまります。

関連情報: *Anybus CompactCom 40 Software Design Guide (HMSI-216-125)*、「コントロールレジスタ」

G.3.8 ステータスレジスタ、プロセスデータリード領域、メッセージデータリード領域の更新

40シリーズでは、ステータスレジスタ、プロセスデータリード領域、およびメッセージデータリード領域は、ハードウェア(パラレルインターフェース)内で書き込み保護されています。何らかの理由でソフトウェアがこれらの領域に書き込みを行っている場合は、変更が必要です。

HMS Industrial NetworksからリリースされたAnybus CompactCom 30用サンプルドライバ全てに、この違いが当てはまります。

G.4 ネットワーク固有 – EtherNet/IP

G.4.1 ネットワークオブジェクト (03h)

アトリビュート#1、**Network Type** 30シリーズモジュールは、2つのイーサネット冗長化プロトコル、「ビーコンベースDLR」(最高性能)または「アナウンスベースDLR」が適用されます。一方、40シリーズは「ビーコンベースDLR」のみ適用されます。これらのバージョン間でNetwork Typeの値は異なります。

値	Network Type	Anybus CompactCom 製品
0085h	EtherNet/IP、DLRなし	30シリーズ 1ポート
009Ch	EtherNet/IP、Announce Based DLR	30シリーズ 2ポート
009Bh	EtherNet/IP、Beacon Based DLR	30シリーズおよび40シリーズ
00ABh	EtherNet/IP、Beacon Based DLR + IIoT	40シリーズ

G.4.2 EtherNet/IPホストオブジェクト(F8h)

アトリビュート	デフォルト	Anybus CompactCom 製品	コメント
#2, Device Type	0000h	30シリーズ、EtherNet/IP、DLRなし	アトリビュートがホストアプリケーションに実装されている場合、デフォルト値が無効にされ、30シリーズと40シリーズの間に違いがなくなります。アトリビュートが実装されていない場合、デフォルト値が使用されます。
	0000h	30シリーズ、EtherNet/IP、Announce Based DLR	
	002Bh	30シリーズ、EtherNet/IP、Beacon Based DLR	
	002Bh	40シリーズ、EtherNet/IP、Beacon Based DLR	
#3, Product Code	0063h	30シリーズ、EtherNet/IP、DLRなし	アトリビュートがホストアプリケーションに実装されている場合、デフォルト値が無効にされ、30シリーズと40シリーズの間に違いがなくなります。アトリビュートが実装されていない場合、デフォルト値が使用されます。
	002Eh	30シリーズ、EtherNet/IP、Announce Based DLR	
	0036h	30シリーズ、EtherNet/IP、Beacon Based DLR	
	0037h	40シリーズ、EtherNet/IP、Beacon Based DLR	
#6, Product Name	Anybus-CC EtherNet/IP	30シリーズ、EtherNet/IP、DLRなし	アトリビュートがホストアプリケーションに実装されている場合、デフォルト値が無効にされ、30シリーズと40シリーズの間に違いがなくなります。アトリビュートが実装されていない場合、デフォルト値が使用されます。
	CompactCom EtherNet/IP(TM) 2P	30シリーズ、EtherNet/IP、Announce Based DLR	
	Anybus-CC EIP (2-Port) BB DLR	30シリーズ、EtherNet/IP、Beacon Based DLR	
	Anybus CompactCom 40 EtherNet/IP(TM)	40シリーズ、EtherNet/IP、Beacon Based DLR	
アトリビュート #27、Producing Instance Map	コメント参照		40シリーズでアトリビュートを削除 (30シリーズEtherNet/IP Beacon Based DLRでのみ利用可能)。CompactComがこのアトリビュートを要求することはありません。アセンブリマッピングオブジェクト (EBh) の機能に置き換えられました。このアトリビュートを使用する場合、アセンブリマッピングオブジェクトが代わりに実装されなければなりません。
アトリビュート #28、Consuming Instance Map	コメント参照		40シリーズでアトリビュートを削除 (30シリーズEtherNet/IP Beacon Based DLRでのみ利用可能)。CompactComがこのアトリビュートを要求することはありません。アセンブリマッピングオブジェクト (EBh) の機能に置き換えられました。このアトリビュートを使用する場合、アセンブリマッピングオブジェクトが代わりに実装されなければなりません。

EtherNet/IP 機能

最大メッセージコネクション数	Class3コネクションは、30シリーズでは最大16、40シリーズでは最大6つ同時に接続できます。 ホストアプリケーションに変更は必要ありません。
EtherNet/IP Encapsulation セッション	Encapsulationセッションは、30シリーズでは最大48、40シリーズでは最大15同時に使用可能です。 ホストアプリケーションに変更は必要ありません。

G.4.3 EDSファイル (設定ツールが使用する電子データファイル)

EDSファイルジェネレータ

自動EDSファイル生成ツールであるEDSジェネレータは下記仕様を含めEDSを作成します。EDSジェネレータは、40シリーズのバージョン1.30以降でのみ動作します。

ジェネレータはwww.anybus.com/starterkit40からダウンロードできます。

キーワード

次のキーワードは30シリーズと40シリーズで異なります。EDSジェネレータにはこの変更が反映されています。

キーワード	コメント
Capacity->MaxCIPConnections	40シリーズでは削除 - 代替: MaxMsgConnectionsおよびMaxIOConnections (下記参照)
Capacity->MaxMsgConnections	40シリーズにおける新キーワード、値: 6
Capacity->MaxIOConnections	40シリーズにおける新キーワード、値: 4

H ライセンス情報

lwIPはBSDからライセンスを受けています。

Copyright (c) 2001-2004 Swedish Institute of Computer Science.

All rights reserved.

ソースおよびバイナリ形式での再配布および使用は、変更の有無にかかわらず、以下の条件を満たす場合に許可されます。

1. ソースコードを再配布する場合は、上記の著作権表示、本条件リスト、および以下の免責表示を含める必要があります。
2. バイナリ形式で再配布する場合は、上記の著作権表示、本条件リスト、および下記の免責表示を、配布に際して提供する関連文書および資料に記載する必要があります。
3. 書面による事前の許可なしに、著作権保持者または貢献者の名前を、本ソフトウェアの二次創作品の宣伝または販売促進のために使用することはできません。

本ソフトウェアは執筆者によって「現状のまま」提供されるものとし、明示黙示を問わず、商品性の黙示保証および特定目的に対する適合性を含め、これらに限定されず何の保証もなされません。執筆者はいかなる場合にも、本ソフトウェアの使用の結果生じた直接的、間接的、付随的、特別、懲罰的、または結果的な損害 (代替商品またはサービスの調達、使用機会の損失、データの喪失、利益の逸失、または事業の中断を含むが、これらに限定されない) に対して、事由や損害発生の原因のいかんを問わず、かつ責任の根拠が契約、厳格責任、不法行為 (過失その他を含む) であるかを問わず、仮にそのような損害が発生する可能性を知らされていたとしても、一切責任を負いません。

印刷形式指定のルーチン

Copyright (C) 2002 Michael Ringgaard. All rights reserved.

ソースおよびバイナリ形式での再配布および使用は、変更の有無にかかわらず、以下の条件を満たす場合に許可されます。

1. ソースコードを再配布する場合は、上記の著作権表示、本条件リスト、および以下の免責表示を含める必要があります。
2. バイナリ形式で再配布する場合は、上記の著作権表示、本条件リスト、および下記の免責表示を、配布に際して提供する関連文書および資料に記載する必要があります。
3. 書面による事前の許可なしに、プロジェクト名または貢献者の名前を、本ソフトウェアから派生した製品の宣伝または販売促進のために使用することはできません。

本ソフトウェアは著作権保持者および貢献者によって「現状のまま」提供されるものとし、明示黙示を問わず、商品性の黙示保証および特定目的に対する適合性を含め、これらに限定されず何の保証もなされません。著作権所有者または貢献者はいかなる場合にも、本ソフトウェアの使用の結果生じた直接的、間接的、付随的、特別、懲罰的、または結果的な損害 (代替商品またはサービスの調達、使用機会の損失、データの喪失、利益の逸失、または事業の中断を含むが、これらに限定されない) に対して、事由や損害発生の原因のいかんを問わず、かつ責任の根拠が契約、厳格責任、不法行為 (過失その他を含む) であるかを問わず、仮にそのような損害が発生する可能性を知らされていたとしても、一切責任を負いません。

Copyright (c) 2002 Florian Schulze.

無断複写・複製・転載禁止。

ソースおよびバイナリ形式での再配布および使用は、変更の有無にかかわらず、以下の条件を満たす場合に許可されます。

1. ソースコードを再配布する場合は、上記の著作権表示、本条件リスト、および以下の免責表示を含める必要があります。
2. バイナリ形式で再配布する場合は、上記の著作権表示、本条件リスト、および下記の免責表示を、配布に際して提供する関連文書および資料に記載する必要があります。
3. 書面による事前の許可なしに、執筆者の名前または貢献者の名前を、本ソフトウェアから派生した製品の宣伝または販売促進のために使用することはできません。

本ソフトウェアは執筆者および貢献者によって「現状のまま」提供されるものとし、明示黙示を問わず、商品性の黙示保証および特定目的に対する適合性を含め、これらに限定されず何の保証もなされません。執筆者または貢献者はいかなる場合にも、本ソフトウェアの使用の結果生じた直接的、間接的、付随的、特別、懲罰的、または結果的な損害（代替商品またはサービスの調達、使用機会の損失、データの喪失、利益の逸失、または事業の中断を含むが、これらに限定されない）に対して、事由や損害発生の原因のいかんを問わず、かつ責任の根拠が契約、厳格責任、不法行為（過失その他を含む）であるかを問わず、仮にそのような損害が発生する可能性を知らされていたとしても、一切責任を負いません。

ftpd.c - このファイルはlwIPのFTPデーモンに属します

FatFs - FATファイルシステムモジュールR0.09b (C)ChaN、2013

FatFsモジュールは、小規模な組み込みシステム用のジェネリックFATファイルシステムモジュールです。これは、次のようなライセンスポリシー規約に基づき、教育・研究・商業開発のために公開されているフリーソフトウェアです。

Copyright (C) 2013, ChaN. 無断複写・複製・転載禁止。

FatFsモジュールはフリーソフトウェアです。保証はありません。使用上の制限はありません。個人の責任において、私用品・非営利品・商品のために使用・変更・再配布することができます。ソースコードを再配布する場合は、上記の著作権表示が含まれている必要があります。

Copyright (c) 2016 The MINIX 3 Project.
無断複写・複製・転載禁止。

ソースおよびバイナリ形式での再配布および使用は、変更の有無にかかわらず、以下の条件を満たす場合に許可されます。

1. ソースコードを再配布する場合は、上記の著作権表示、本条件リスト、および以下の免責表示を含める必要があります。
2. バイナリ形式で再配布する場合は、上記の著作権表示、本条件リスト、および下記の免責表示を、配布に際して提供する関連文書および資料に記載する必要があります。
3. 書面による事前の許可なしに、著作権保持者または貢献者の名前を、本ソフトウェアの二次創作品の宣伝または販売促進のために使用することはできません。

本ソフトウェアは執筆者によって「現状のまま」提供されるものとし、明示黙示を問わず、商品性の黙示保証および特定目的に対する適合性を含め、これらに限定されず何の保証もなされません。執筆者はいかなる場合にも、本ソフトウェアの使用の結果生じた直接的、間接的、付随的、特別、懲罰的、または結果的な損害（代替商品またはサービスの調達、使用機会の損失、データの喪失、利益の逸失、または事業の中断を含むが、これらに限定されない）に対して、事由や損害発生の原因のいかんを問わず、かつ責任の根拠が契約、厳格責任、不法行為（過失その他を含む）であるかを問わず、仮にそのような損害が発生する可能性を知らされていたとしても、一切責任を負いません。

執筆者: David van Moolenbroek <david@minix3.org>

MD5ルーチン

Copyright (C) 1999, 2000, 2002 Aladdin Enterprises. 無断複写・複製・転載禁止。

このソフトウェアは「現状のまま」提供されるものとし、明示的にも黙示的にも保証はありません。このソフトウェアの使用に基づいて発生した損害に対して、いかなる場合にも執筆者は責任を負いません。任意の者は商用アプリケーションを含む任意の目的のためにこのソフトウェアを使用し、次の制約の下で自由に変更し再配布することができます。

1. このソフトウェアの出所は虚偽表示されてはならず、あなたが純正ソフトウェアの作成者であると偽る表示を行うことはできません。このソフトウェアを製品に使用する場合、製品ドキュメンテーションに謝辞を含めることが望ましいあり方ですが、義務付けられているわけではありません。
2. 改変されたソースバージョンは明白にその旨を表記し、純正ソフトウェアであるかのような虚偽表示を行うことはできません。
3. この通知はどのソースバージョンからも削除や改変することもできません。

L.Peter Deutsch

ghost@aladdin.com

Copyright 2013 jQuery Foundationおよびその他の寄与者

<http://jquery.com/>

本ソフトウェアおよび関連文書ファイル(以下「ソフトウェア」)のコピーを入手した人は、本ソフトウェアを無制限に扱うことができ、これには、本ソフトウェアの使用、コピー、修正、結合、出版、配布、サブライセンス、および/またはコピーの販売する権利を含むが、本ソフトウェアを提供された人がそうすることの許可を、以下の条件のもとで無償で認められます。

上記の著作権表記とこの許諾通知書がすべてのコピーまたはソフトウェアの大半の部分に記載されること。

本ソフトウェアは「現状のまま」提供されるものとし、明示黙示を問わず、商品性の保証、特定目的に対する適合性、他者の権利の非侵害性を含め、これらに限定されず何の保証もなされません。執筆者または著作権所有者はいかなる場合にも、ソフトウェア自体、またはソフトウェアの使用やソフトウェアのその他の取扱いに関連して発生した賠償請求、損害、またはその他の法的責任について、契約行為、不正行為、それ以外によるかを問わず、一切の責任を負いません。

rsvp.js

Copyright (c) 2013 Yehuda Katz, Tom Daleおよび貢献者

本ソフトウェアおよび関連文書ファイル(以下「ソフトウェア」)のコピーを入手した人は、本ソフトウェアを無制限に扱うことができ、これには、本ソフトウェアの使用、コピー、修正、結合、出版、配布、サブライセンス、および/またはコピーの販売する権利を含むが、本ソフトウェアを提供された人がそうすることの許可を、以下の条件のもとで無償で認められます。

上記の著作権表記とこの許諾通知書がすべてのコピーまたはソフトウェアの大半の部分に記載されること。

本ソフトウェアは「現状のまま」提供されるものとし、明示黙示を問わず、商品性の保証、特定目的に対する適合性、他者の権利の非侵害性を含め、これらに限定されず何の保証もなされません。執筆者または著作権所有者はいかなる場合にも、ソフトウェア自体、またはソフトウェアの使用やソフトウェアのその他の取扱いに関連して発生した賠償請求、損害、またはその他の法的責任について、契約行為、不正行為、それ以外によるかを問わず、一切の責任を負いません。

libb (big.js)

MIT Expat Licence.

Copyright (c) 2012 Michael Mclaughlin

本ソフトウェアおよび関連文書ファイル(以下ソフトウェア)のコピーを入手した人は、本ソフトウェアを無制限に扱うことができ、これには、本ソフトウェアの使用、コピー、修正、結合、出版、配布、サブライセンス、および/またはコピーの販売する権利を含むが、本ソフトウェアを提供された人がそうすることの許可を、以下の条件のもとで無償で認められます。

上記の著作権表記とこの許諾通知書がすべてのコピーまたはソフトウェアの大半の部分に記載されること。

本ソフトウェアは「現状のまま」提供されるものとし、明示黙示を問わず、商品性の保証、特定目的に対する適合性、他者の権利の非侵害性を含め、これらに限定されず何の保証もなされません。執筆者または著作権所有者はいかなる場合にも、ソフトウェア自体、またはソフトウェアの使用やソフトウェアのその他の取扱いに関連して発生した賠償請求、損害、またはその他の法的責任について、契約行為、不正行為、それ以外によるかを問わず、一切の責任を負いません。

「inih」ライブラリは修正BSDライセンスを受けて配布されています。

Copyright (c) 2009, Ben Hoyt

無断複写・複製・転載禁止。

ソースおよびバイナリ形式での再配布および使用は、変更の有無にかかわらず、以下の条件を満たす場合に許可されます。

* ソースコードを再配布する場合は、上記の著作権表示、本条件リスト、および以下の免責表示を含める必要があります。

* バイナリ形式で再配布する場合は、上記の著作権表示、本条件リスト、および下記の免責表示を、配布に際して提供する関連文書および資料に記載する必要があります。
* 書面による事前の許可なしに、Ben Hoytという名前または貢献者の名前を、本ソフトウェアから派生した製品の宣伝または販売促進のために使用することはできません。

本ソフトウェアはBEN HOYTによって「現状のまま」提供されるものとし、明示黙示を問わず、商品性の黙示保証および特定目的に対する適合性を含め、何の保証もなされません。BEN HOYTはいかなる場合にも、本ソフトウェアの使用の結果生じた直接的、間接的、付随的、特別、懲罰的、または結果的な損害(代替商品またはサービスの調達、使用機会の損失、データの喪失、利益の逸失、または事業の中断を含むが、これらに限定されない)に対して、事由や損害発生の原因のいかんを問わず、かつ責任の根拠が契約、厳格責任、不法行為(過失その他を含む)であるかを問わず、仮にそのような損害が発生する可能性を知らされていたとしても、一切責任を負いません。

open62541はMozilla Public License v2.0からライセンスを受けています

このソースコードフォームはMozilla Public License、v.2.0の利用規約に従うものとし、このファイルと共にMPLのコピーが配布されなかった場合は、<http://mozilla.org/MPL/2.0/>で入手することができます。

カスタマイズした変更の取得については、foss@anybus.comまでお問い合わせください。

muslは、その全体が以下の標準MITライセンスからライセンスを受けています。

Copyright © 2005-2014 Rich Felker, et al.

本ソフトウェアおよび関連文書ファイル(以下「ソフトウェア」)のコピーを入手した人は、本ソフトウェアを無制限に扱うことができ、これには、本ソフトウェアの使用、コピー、修正、結合、出版、配布、サブライセンス、および/またはコピーの販売する権利を含むが、本ソフトウェアを提供された人がその許可を、以下の条件のもとで無償で認められます。

上記の著作権表記とこの許諾通知書がすべてのコピーまたはソフトウェアの大半の部分に記載されること。

本ソフトウェアは「現状のまま」提供されるものとし、明示黙示を問わず、商品性の保証、特定目的に対する適合性、他者の権利の非侵害性を含め、これらに限定されず何の保証もなされません。執筆者または著作権所有者はいかなる場合にも、ソフトウェア自体、またはソフトウェアの使用やソフトウェアのその他の取扱いに関連して発生した賠償請求、損害、またはその他の法的責任について、契約行為、不正行為、それ以外によるかを問わず、一切の責任を負いません。

C用PCG乱数の生成

Copyright 2014 Melissa O'Neill <oneill@pcg-random.org>

Apache License、Version 2.0(以下「ライセンス」)からライセンスを受けています。本ライセンスに準拠する場合を除き、このファイルを使用することはできません。ライセンスのコピーは以下で入手できます。

<http://www.apache.org/licenses/LICENSE-2.0>

適用される法律による要求または書面による同意がない限り、本ライセンスのもとに配布されるソフトウェアは「現状のまま」配布され、明示・黙示を問わずいかなる類の保証も条件もないものとし、本ライセンスのもとで特定の言語について定められている許可および制限事項については、ライセンスを参照してください。

PCG乱数生成スキームの詳細、ならびにライセンスおよびその他のライセンスオプションについては、以下をご覧ください。

<http://www.pcg-random.org>

queue.h

Copyright (c) 1991, 1993

カリフォルニア大学理事無断複写・複製・転載禁止。

ソースおよびバイナリ形式での再配布および使用は、変更の有無にかかわらず、以下の条件を満たす場合に許可されます。

1. ソースコードを再配布する場合は、上記の著作権表示、本条件リスト、および以下の免責表示を含める必要があります。
2. バイナリ形式で再配布する場合は、上記の著作権表示、本条件リスト、および下記の免責表示を、配布に際して提供する関連文書および資料に記載する必要があります。
3. 書面による事前の許可なしに、大学名または貢献者の名前を、本ソフトウェアから派生した製品の宣伝または販売促進のために使用することはできません。

本ソフトウェアは理事および貢献者によって「現状のまま」提供されるものとし、明示黙示を問わず、商品性の黙示保証および特定目的に対する適合性を含め、これらに限定されず何の保証もなされません。理事または貢献者はいかなる場合にも、本ソフトウェアの使用の結果生じた直接的、間接的、付随的、特別、懲罰的、または結果的な損害（代替商品またはサービスの調達、使用機会の損失、データの喪失、利益の逸失、または事業の中断を含むが、これらに限定されない）に対して、事由や損害発生の原因のいかんを問わず、かつ責任の根拠が契約、厳格責任、不法行為（過失その他を含む）であるかを問わず、仮にそのような損害が発生する可能性を知らされていたとしても、一切責任を負いません。

@(#)queue.h 8.5 (Berkeley) 8/20/94

Format - lightweight string formatting library.

Copyright (C) 2010-2013, Neil Johnson

無断複写・複製・転載禁止。

ソースおよびバイナリ形式での再配布および使用は、変更の有無にかかわらず、以下の条件を満たす場合に許可されます。

- * ソースコードを再配布する場合は、上記の著作権表示、本条件リスト、および以下の免責表示を含める必要があります。
- * バイナリ形式で再配布する場合は、上記の著作権表示、本条件リスト、および下記の免責表示を、配布に際して提供する関連文書および資料に記載する必要があります。
- * 書面による事前の許可なしに、主導者の名前または貢献者の名前を、本ソフトウェアから派生した製品の宣伝または販売促進のために使用することはできません。

本ソフトウェアは著作権保持者および貢献者によって「現状のまま」提供されるものとし、明示黙示を問わず、商品性の黙示保証および特定目的に対する適合性を含め、これらに限定されず何の保証もなされません。著作権所有者または貢献者はいかなる場合にも、本ソフトウェアの使用の結果生じた直接的、間接的、付随的、特別、懲罰的、または結果的な損害（代替商品またはサービスの調達、使用機会の損失、データの喪失、利益の逸失、または事業の中断を含むが、これらに限定されない）に対して、事由や損害発生の原因のいかんを問わず、かつ責任の根拠が契約、厳格責任、不法行為（過失その他を含む）であるかを問わず、仮にそのような損害が発生する可能性を知らされていたとしても、一切責任を負いません。

