

Anybus[®] CompactCom[™] 40

EtherCAT[®]

NETWORK GUIDE

SCM-1202-034 1.9 ja-JP 日本語

必ずお読みください

責任の範囲

本ドキュメントは細心の注意を払って作成されています。誤字や脱字があればHMS Industrial Networks ABまでご連絡ください。本ドキュメントに記載されているデータや図表は、何ら拘束力を持ちません。HMS Industrial Networks ABは、製品開発へ継続的に取り組むという自社ポリシーに基づき、製品に変更を加える権利を留保します。本ドキュメントの内容は予告なく変更される場合があります。また、本ドキュメントの内容はHMS Industrial Networks ABによる何らかの保証を表明するものではありません。HMS Industrial Networks AB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。

本製品は様々な用途に応用可能です。本装置の使用者は、必要なあらゆる手段を通じて、本装置の用途が適用される法令、規則、規約、規格の定める性能・安全性に関する要件をすべて満たしていることを検証しなければならないものとします。

HMS Industrial Networks AB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やタイミング、機能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。

本ドキュメントの例および図表は、説明のみを目的として使用されています。本製品の個々の使用においては様々なバリエーションや要件が存在するため、本ドキュメントの例や図表に基づいて本製品を使用したことに関して、HMS Industrial Networks ABは一切の責任を負いません。

知的所有権

本ドキュメントに記載されている製品に組み込まれた技術に関連する知的所有権は、HMS Industrial Networks ABに帰属します。この知的所有権には、米国およびその他の国における特許および出願中の特許出願が含まれる場合があります。

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	表記と用語	6
1.5	ドキュメント固有の表記規則	6
1.6	商標について	7
2	Anybus CompactCom 40 EtherCAT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	EtherCAT実装詳細	14
3.3	CANopen over EtherCATの実装詳細	16
3.4	データ交換	17
3.5	ファイルシステム	18
3.6	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	20
3.7	ネットワークのリセット処理	21
3.8	設定ステーションエイリアス (ノードアドレス)	22
3.9	Device ID	22
3.10	モジュラーデバイスプロファイル	22
4	オブジェクトディクショナリ (CANopen over EtherCAT)	23
4.1	標準オブジェクト	23
4.2	メーカー及び、プロファイル固有オブジェクト	28
5	Anybus モジュールオブジェクト	34
5.1	基本情報	34
5.2	Anybusオブジェクト (01h)	35
5.3	診断オブジェクト (02h)	36
5.4	ネットワーク オブジェクト (03h)	38
5.5	ネットワークコンフィグレーションオブジェクト (04h)	40
5.6	ソケットインターフェースオブジェクト (07h)	45
5.7	SMTPクライアントオブジェクト (09h)	62
5.8	ネットワークイーサネットオブジェクト (0Ch)	67
5.9	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)	69

6	ホストアプリケーションオブジェクト.....	75
6.1	基本情報	75
6.2	ファンクショナルセーフティオブジェクト (E8h)	76
6.3	アセンブリマッピングオブジェクト (EBh)	78
6.4	SYNCオブジェクト (EEh)	79
6.5	EtherCATオブジェクト (F5h)	81
6.6	Ethernetホストオブジェクト (F9h)	88
7	Webサーバ.....	92
7.1	基本情報	92
7.2	デフォルト Webページ	92
7.3	サーバコンフィグレーション	95
8	FTPサーバ.....	98
8.1	基本情報	98
8.2	ユーザーアカウント	98
8.3	セッション例.....	99
9	電子メールクライアント.....	100
9.1	基本情報	100
9.2	電子メールメッセージを送る方法.....	100
10	Server Side Include (SSI)	101
10.1	基本情報	101
10.2	Include File	101
10.3	コマンドの機能	101
10.4	引数関数	115
10.5	SSI出力の設定	119
11	JSON	120
11.1	基本情報	120
11.2	JSONオブジェクト	121
11.3	例	140
A	機能の分類.....	141
A.1	基本	141
A.2	拡張	141

B	実装詳細.....	142
B.1	SUPビットの定義	142
B.2	Anybusステートマシン	142
B.3	アプリケーションステータスレジスタ	143
B.4	Application Watchdog Timeout Handling.....	143
C	技術仕様.....	144
C.1	前面図.....	144
C.2	ファンクショナルアース (FE) の要件	145
C.3	電源	146
C.4	環境仕様	146
C.5	EMC準拠.....	146
D	タイミングと性能.....	147
D.1	基本情報	147
D.2	内部タイミング	147
E	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)	149
E.1	概要	149
E.2	操作	149
F	後方互換性.....	150
F.1	初期段階における注意事項.....	150
F.2	ハードウェアの互換性	151
F.3	ソフトウェア全般	155
F.4	ネットワーク固有 — EtherCAT.....	157
G	著作権表示.....	159

このページは意図的に空白になっています

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 EtherCAT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 EtherCAT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、EtherCAT固有の高度な機能を使用する場合は、EtherCATネットワーキングの内部詳細、および公式のEtherCAT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、EtherCAT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
IEC 61158-6	IEC	
CiA Draft Standard 301 v4.02	CAN in Automation	

1.3 ドキュメント更新履歴

バージョン	日付	説明
1.1	2017/01/19	FMからDOXに、文書番号をHMSI-27-220からSCM-1202-027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
1.2	2017/06/15	Get_Object_Descriptionサービスとアトリビュート20、21、22をEtherCATオブジェクトに追加 モジュラーデバイスプロファイル、オブジェクトエントリのセクションに注記を追加 ウォッチドッグ機能に、PDIウォッチドッグに対応していることを記載 オブジェクトディクショナリのセクションに情報を追加 ESIジェネレーターに関する情報をESIのセクションに追加 ERR LEDに関する説明に、ウォッチドッグ機能セクションへの参照を追加 小規模な修正
1.3	2017/07/11	後方互換性に関するAppendix追加
1.4	2017/09/29	セーフティオブジェクト (11h、E8h) およびFSoE (Fail Safe over EtherCAT) に関する情報を追加
1.5	2017/12/04	ネットワークイーサネットオブジェクト (0Ch) を更新
1.6	2018/01/11	商標に関する情報を更新
1.7	2018/01/17	商標に関する情報を更新
1.8	2018/05/25	商標に関する情報を更新 ネットワークデータ型について更新 EtherCATホストオブジェクトを更新 小規模な修正
1.9	2018/11/01	EtherCATオブジェクト (F5h) にインスタンスアトリビュート#27を追加 ネットワークのリセットに関する説明を更新

1.4 表記と用語

順番通りに実行されなくてはならない指示については、番号の付いたリストが使用されます。

1. まずこれを行います
2. その後これを行います

順番のない指示については、番号付けのないリスト (箇条書き) が使用されます。

- 項目化された情報
- 任意の順序で実行できる指示

アクションと結果が対になる指示については、以下のように表記します。

- ▶ このアクションは...
- この結果につながります

Bold typefaceはコネクタ、スイッチなどハードウェア上のインタラクティブな部品、またはグラフィックユーザーインターフェース上のメニューやボタンを示します。

等幅フォントはプログラムコードやコンフィギュレーションスクリプトなどのデータ入出力表示などに使用されます。

これはこの文書内の相互参照です: [表記と用語, ページ 6](#)

これは、外部リンク (URL) です: www.hms-networks.com



これは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です。



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、この指示には従わなければなりません。



注意

個人の負傷のリスクを避けるため、この指示には従わなければなりません。



警告

死亡または重篤な障害のリスクを避けるため、この指示には従わなければなりません。

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 商標について

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



EtherCAT® は、ドイツの Beckhoff Automation GmbH よりライセンスを受けた登録商標および特許技術です。



Safety over EtherCAT® は、ドイツの Beckhoff Automation GmbH よりライセンスを受けた登録商標および特許技術です。

その他の商標は、各所有者に帰属します。

2 Anybus CompactCom 40 EtherCATについて

2.1 概要

Anybus CompactCom 40 EtherCAT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、EtherCAT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、モジュールに提供される機能を活用して、そのタイプを問わず、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。

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

- CANopen over EtherCAT (CoE)
 - Complete Accessに対応
- モジュラーデバイスプロファイルに対応
- イーサネットコネクタまたはM12コネクタ
- DS301に準拠
- ガルバニック絶縁されたバス回路
- ネットワークIDのカスタマイズ
- EMCYに関するサポート
- メーカー固有オブジェクトおよびデバイスプロファイル固有オブジェクトとして最大57343個のADIにネットワークからアクセス可能 (一般モード)
- メーカー固有オブジェクトおよびデバイスプロファイル固有オブジェクトとして最大16383個のADIにネットワークからアクセス可能 (モジュラーデバイスプロファイルを有効にした場合)
- 各方向毎に、最大1486バイトの高速サイクリックI/O
- HMSにより提供されたEtherCATスレーブインターフェースファイル
- ディストリビューテッドクロックを使用したSync0機能に対応
- Ethernet over EtherCAT (EoE)
- コンテンツのカスタム化が可能なWebサーバ
- FTPサーバ
- 電子メールクライアント
- Server Side Include (SSI) 機能
- JSON機能
- ブラックチャネルインターフェース。これでFail Safe over EtherCAT (FSoE) に対応する透過チャネルを提供します。
- File access over EtherCAT (FoE)
- ネットワークからのプロセスデータの再マッピングをサポート
- ネットワークサイクルタイムを100μsに低減
- DS402デバイスプロファイル、Semiデバイスプロファイル、およびその他のデバイスプロファイルを実装可能



TwinCAT 3ツール (または2.11以降のバージョン) が使用されている場合、ツールの制限により、最大プロセスデータ量は1473バイトとなります。

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 EtherCAT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、EtherCAT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。

- ネットワークからアクセスできるADIの最大インスタンス番号は57343 (DFFFh) です。モジュラーデバイスプロファイルが実装され、そのプロファイルが実行されている場合、インスタンス番号は16383 (3FFFh) に制限されます。
- ADIをプロセスデータにマッピングする場合、どちらの方向においても、1486個の要素または1486バイトのいずれか先に到達した上限を超えてマッピングすることはできません。
- Anybusのコンセプトは柔軟であるため、アプリケーションからEtherCATの動作を変更した結果、標準のEtherCATスレーブ情報ファイルと矛盾したり、ネットワークの認定が無効となったりする場合があります。最終製品の開発者は、ネットワーク認定と相互運用性に関し、それらの製品が各自の要求やポリシーを確実に満たすようにしてください。
- EtherCATの高度な機能を使用するには、EtherCATネットワーキングの内部詳細、および公式のEtherCAT規格に関する十分な知識が必要です。本製品の使用者は、EtherCAT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本製品を使用してください。



TwinCAT 3ツール (または2.11以降のバージョン) が使用されている場合、ツールの制限により、最大プロセスデータ量は1473バイトとなります。

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 EtherCATスレーブインターフェース (ESI) ファイル

EtherCAT上の各機器は、XML形式のEtherCATスレーブインターフェース (ESI) ファイルと関連付けられています。このファイルには、機器やその機能の説明が記述されています。

相互運用性を確保し、エンドユーザーに煩雑な作業を行わせないようにするため、製品の最終的な実装形態に合わせたカスタムESIファイルを作成することを強く推奨します。ESIファイルの作成が容易に行えるよう、HMSはHMS EtherCAT ESI Generatorというツールを提供しています。このツールはwww.anybus.comのAnybus CompactCom 40 EtherCAT製品ページから無料でダウンロードできます。

EtherCAT Technology Group (ETG) は、最終製品のベンダーに応じてベンダーIDを変更するよう要求しています。以下のシナリオでは、EtherCATスレーブインターフェースファイルに追加の変更が必要となる場合があります。

- カスタムの製品コードを使用している。
- 独自のベンダーIDを使用している。
- 製品リビジョンを変更する。
- ホストアプリケーションがRemap_ADIコマンドをサポートしている。
- Ethernet over EtherCAT (EoE) を使用している。
- アプリケーションの応答時間が遅い。HMSが提供する標準のESIファイルに従う為に、1ms以内に明示的な要求を処理する必要があります。これは、数多くのI/Oを持つ低速なシリアルリンクに対しては十分ではありません (この場合、ファイルに記述されているメールボックスのタイムアウト値を状況に応じて増やす必要があります)。



なお、標準のESIファイルを変更した場合、必須のカスタムベンダーIDの他に、カスタムの製品コードを使用する必要があります。

3.1.3 デバイスIdentity

標準の実装形態 (ネットワーク固有の機能が実装されていない状態) では、本モジュールは以下のIdentity情報を持つHMSの標準の機器とみなされます。

オブジェクトエントリ	値
Vendor ID	E000 0018h (HMS Industrial NetworksのセカンダリベンダーID。最終製品のベンダーIDに置き換えること)
Product Code	0000 0036h (Anybus CompactCom 40 EtherCAT)
Device Name	Anybus CompactCom 40 EtherCAT
Serial Number	(製造時に割り当て)

EtherCAT Object (F5h) のサポートを実装する事で、標準のAnybus機器ではなく、ベンダー固有の機器として扱われるように本モジュールをカスタマイズできます。最終製品がETGのコンFORMANCEテストに合格して認定を受けるには、別のベンダーIDをETGに要求する必要があります。

下記も参照してください。

- [EtherCATオブジェクト \(F5h \)](#), ページ 81

3.1.4 File Access over EtherCAT (FoE)

本モジュールは、ファームウェアファイルをクライアントマシンからサーバにダウンロードできるようにFile Access over EtherCAT (FoE) をサポートしています。拡張子がまたは

FoEを使用してダウンロードしたファームウェアファイルの更新が保留になっている場合、拡張子が

Anybus IPはFoEに対応していません。

3.1.5 Fail Safe over EtherCAT (FSoE)

Anybus CompactCom 40 EtherCATはFSoEに対応しています。このプロファイルにより、ブラックチャネルインターフェースからデータを送信することが可能になります。IXXAT Safe T100などのアドオンセーフティモジュールを使ってEtherCAT上でのセーフチャネルが確立されます。アプリケーションがFSoEに対応できるためには、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) の実装が必要になります。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用に使用されます。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用に実装されます。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 69
- [ファンクショナルセーフティオブジェクト \(E8h \)](#), ページ 76

3.1.6 Ethernet over EtherCAT (EoE)

モジュールは、EtherCATスレーブとやり取りするための非EtherCATイーサネットフレームの透過トンネルを、Ethernet over EtherCAT (EoE) を使用してサポートします。

制限付きモードのAnybus IPはEoEに対応していません。

Ethernet over EtherCAT (EoE) を使用する場合、次の機能に対応します。

- コンテンツのカスタム化が可能なWebサーバ
- FTPサーバ
- 電子メールクライアント
- Server Side Include (SSI) 機能
- JSON機能

イーサネットフレームはメールボックス通信に組み込まれているため、通常のイーサネット通信と比較すると性能が低下します。データのスループットは、以下によって左右されます。

- EtherCATのプロセスデータサイクルタイム
- メールボックスのサイズ (バイト数)



Ethernet over EtherCAT (EoE) を使用できるようにするには、Anybus CompactCom 40デバイスにMACアドレスを割り当てる必要があります。古いソフトウェア (2.00より前のソフトウェアバージョン) を含むデバイスをご使用の場合は、イーサネットホストオブジェクトでMACアドレスを手動設定して、Ethernet over EtherCAT (EoE) を使用できるようにする必要があります。

Ethernet over EtherCAT (EoE) のサポートをESIファイル内で示す方法または削除する方法については、以下を参照してください。

Ethernet over EtherCAT (EoE) をサポートするためには、<Mailbox>要素が次のようである必要があります。

```
<Mailbox DataLinkLayer="1">  
  <EoE IP="0" MAC="0" TimeStamp="0" />  
  <CoE SdoInfo="1" CompleteAccess="1" PdoAssign="0" PdoConfig="0"  
    PdoUpload="1"/>  
  <FoE/>  
</Mailbox>
```

Ethernet over EtherCAT (EoE) への対応を削除するには、<Mailbox>要素が次のようである必要があります。

```
<Mailbox DataLinkLayer="1">  
  <CoE SdoInfo="1" CompleteAccess="1" PdoAssign="0" PdoConfig="0"  
    PdoUpload="1"/>  
  <FoE/>  
</Mailbox>
```

3.2 EtherCAT実装詳細

3.2.1 基本情報

本モジュールは、以下の基本プロパティを持つ完全なEtherCATスレーブとして機能します。

アプリケーション層：	CANopen over EtherCAT
FMMU	4
シンクマネージャ	4
RAMサイズ：	16 kバイト

下記も参照してください。

- [CANopen over EtherCATの実装詳細, ページ 16](#)

3.2.2 EtherCATの同期

EtherCATの同期およびジッタ精度は以下の要因によって異なる場合があります。

- マスターが同期フレームを送出する頻度
- 環境温度の変動 (影響大)
- EtherCATスレーブ機器の実装
- スレーブ機器で使用されているイーサネット物理層 (RJ45、E-Busなど)

Anybus CompactCom 40 EtherCATモジュールはどのタイプでも、同期精度は1 μ s未満です。RJ45製品では、精度は50 ns前後 (条件が良好な場合)、E-Bus製品では30 ns前後になります。

3.2.3 シンクマネージャ

本モジュールには4つのシンクマネージャが用意されています。

シンクマネージャ0	メールボックスの書き込み転送 (マスターからスレーブ) に使用されます。 本モジュールの書き込みメールボックスはサイズを設定できます。デフォルトサイズは276バイトで、これは255バイトに関連プロトコルヘッダーとパディングを加えたものです。
シンクマネージャ1	メールボックスの読み出し転送 (スレーブからマスター) に使用されます。 本モジュールの読み出しメールボックスはサイズを設定できます。デフォルトサイズは276バイトで、これは255バイトに関連プロトコルヘッダーとパディングを加えたものです。
シンクマネージャ2	RxPDOが格納されます (実際は、シンクマネージャ2にはリードプロセスデータが格納されます)。
シンクマネージャ3	TxPDOが格納されます (実際は、シンクマネージャ3にはライトプロセスデータが格納されます)。

3.2.4 FMMU

4個のFMMUが用意されています。EtherCATマスターは、目的を問わずFMMUを自由に使用できます。

3.2.5 アドレッシングモード

完全なEtherCATとして、モジュールは次のアドレッシングモードに対応しています。

- 位置アドレッシング
- ノードアドレッシング
- 論理アドレッシング

3.2.6 ウォッチドッグ機能

出力I/Oシンクマネージャウォッチドッグ

このウォッチドッグを有効にすると、AnybusモジュールへのPDO通信が監視されます。指定した時間内にマスターがリードプロセスデータを更新しなかった場合、モジュール内でタイムアウト状態が発生し、OPERATIONAL状態からSAFE-OPERATIONAL状態に移行します。このとき、監視ビット (SUP) も影響を受けます。

シンクマネージャウォッチドッグはESIファイルでデフォルトで有効に設定されており、デフォルトの時間間隔は100 msです。

シンクマネージャウォッチドッグは、マスターの設定ツールを使っていつでも無効/有効にできます。

下記も参照してください。

- [SUPビットの定義, ページ 142](#)

PDIウォッチドッグ

PDIウォッチドッグ機能に対応しています。

3.3 CANopen over EtherCATの実装詳細

3.3.1 基本情報

先に述べたように、本モジュールにはCANopen over EtherCATが実装されています。オブジェクトは、DS301通信プロファイルに基づいて実装されています。

下記も参照してください。

- [データ交換, ページ 17](#)
- [オブジェクトディクショナリ \(CANopen over EtherCAT \), ページ 23](#)

3.3.2 実装されているサービス

本モジュールでは、以下のCANopenサービスが実装されています。

サービス	説明
SDO Download Expedited	スレーブに最大4オクテットを書き込む
SDO Download Normal	ネゴシエートされたオクテット数までスレーブに書き込む
Download SDO Segment	オブジェクトのサイズがネゴシエートされたオクテット数を超える場合、追加データを書き込む
SDO Upload Expedited	スレーブから最大4オクテットを読み出す
SDO Upload Normal	ネゴシエートされたオクテット数までスレーブから読み出す
Upload SDO Segment	オブジェクトのサイズがネゴシエートされたオクテット数を超える場合、追加データを読み出す
Abort SDO Transfer	エラー発生時にサーバがサービスを停止
Get OD List	利用可能なインデックスのリストを読み出す
Get Object Description	インデックスの詳細を読み出す
Get Entry Description	サブインデックスの詳細を読み出す
Emergency	予期せぬ状態や診断イベントを報告する

3.4 データ交換

3.4.1 アプリケーションデータ (ADI)

アプリケーションデータインスタンス (ADI) には、メーカー固有の範囲およびプロファイルの範囲 (2001h ~ FFFFh) にある専用のオブジェクトエントリを使用して、ネットワーク経由でアクセスできます。各ノードは、SDO情報プロトコルを使用してADIの名前とデータタイプを取得できます。

下記も参照してください。

- [メーカー及び、プロファイル固有オブジェクト, ページ 28](#)

3.4.2 プロセスデータ

プロセスデータとしてマッピングされたADIは、バス上でプロセスデータオブジェクト (PDO) として周期的に交換されます。実際のPDOマップは、起動時に指定されたプロセスデータマップまたはアプリケーションの実装方法に基づいています。アプリケーションがアプリケーションデータオブジェクトで再マッピングコマンドを実装している場合、実行時にネットワークからPDOマップを変更できます。

本モジュールは最大6個のTPDOと最大6個のRPDOをサポートし、それぞれ最大254個のSDOのマッピングをサポートしています。各SDOは、プロセスデータとしてマッピングされた1個のADI要素に対応しています (複数のADI要素にマッピングすると、複数のSDOにマッピングされます)。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、TPDOとRPDOの数を拡張できます。

PDOマップを変更するには、アセンブリマッピングオブジェクトを使用して再マッピングを行い、起動時に指定されたプロセスデータマップを置き換えます。各PDOは、アセンブリマッピングオブジェクト内のインスタンスによって表されます。モジュールがSafe-Operational状態になると、それらのPDOが再マッピングされます。

モジュラーデバイスオブジェクトが実装されている場合 (すなわち、モジュラーデバイスプロファイルが有効になっている場合)、アセンブリマッピングオブジェクトは無視されます。



できれば、実際のプロセスデータの実装に合わせてEtherCATスレーブ情報ファイルを変更する必要があります。これは一般的な要件ではありませんが、サードパーティ製のマスターとの互換性が高くなります。

下記も参照してください。

- [標準オブジェクト, ページ 23](#)
- [メーカー及び、プロファイル固有オブジェクト, ページ 28](#)
- [アセンブリマッピングオブジェクト \(EBh \), ページ 78](#)
- 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)
- モジュラーデバイスオブジェクト (『Anybus CompactCom 40 Software Design Guide』 を参照)

3.5 ファイルシステム

3.5.1 概要

Anybus CompactCom 40 EtherCAT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。例えば、モジュールのWebページを保持します。
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	ファームウェアディレクトリは、ファームウェアファイルをアップロードできるファームウェア候補領域を示します。

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。

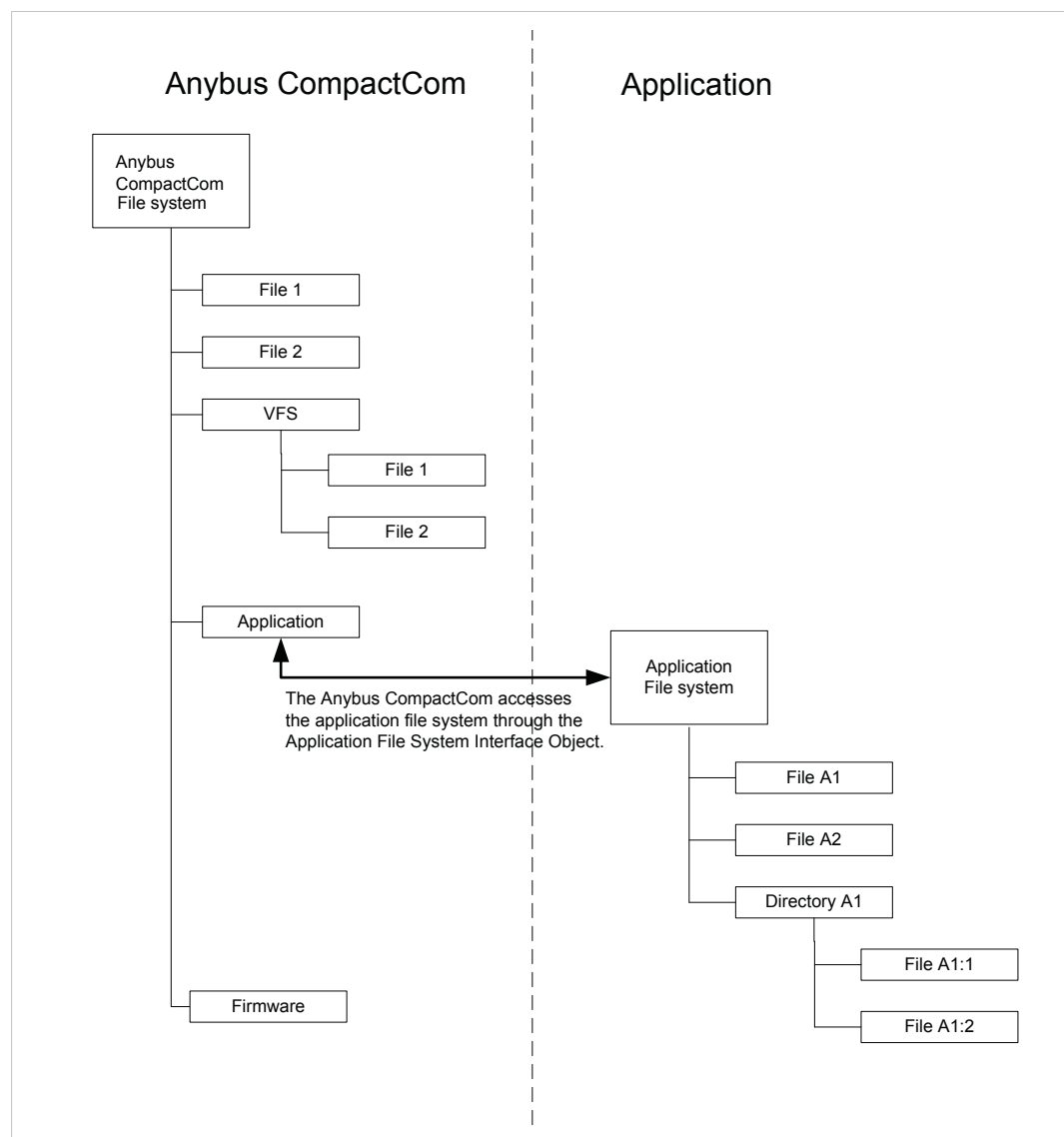


図 1

3.5.2 基本情報

内蔵ファイルシステムは28 Mbの不揮発ストレージになっており、ここへはHTTPやFTPサーバ、Eメールクライアント、ホストアプリケーション (Anybusファイルシステムインターフェースオブジェクト(0Ah)を通して) 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。これより長いファイル名を使用する場合、格納できるディレクトリ/ファイル数は511未満となります。この制限は、ファイルシステム内のその他のディレクトリには適用されません。

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文字と数字です。次の文字は使用できません。 \ / : * ? " < > |
- 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 98](#)
- [Webサーバ, ページ 92](#)
- [電子メールクライアント, ページ 100](#)
- [Server Side Include \(SSI\), ページ 101](#)



ファイルシステムはフラッシュメモリーにあります。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。

以下の演算で1つまたはそれ以上のフラッシュセグメントが消去されます。

- 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

3.5.3 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。これらのファイルの形式は (いくつかの例外を除き) Keyの内容をベースにしており、各Keyに一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。

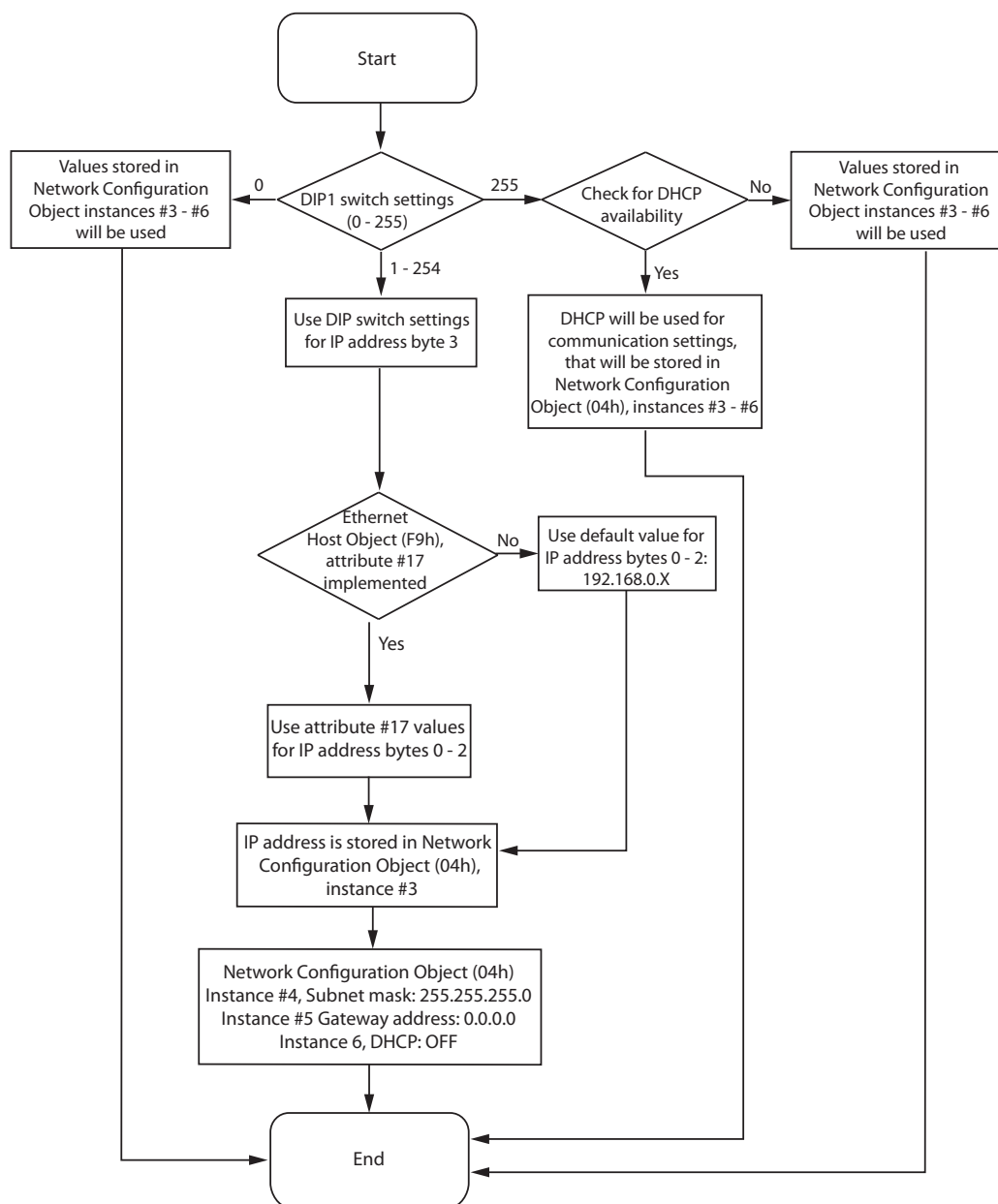
例 1:

```
[Key1]
value of Key1

[Key2]
value of Key2
```

3.6 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 40をスタンドアロンとして使用するときは、IPアドレスを設定するためのアプリケーションがありません。そのため、IPアドレスは、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（イーサネット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。



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 88](#)

[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 40](#)

Anybus CompactCom M40 Hardware Design Guide

3.7 ネットワークのリセット処理

3.7.1 ノードのリセット

FoE (File access over EtherCAT) を使用して有効なファームウェアをダウンロードした場合、Anybus CompactCom 40 EtherCATはBOOTからINITに遷移するときにリセットタイプ00h (パワーオンリセット) をアプリケーションに送信します。リセットを実行できることを確認するため、リセットコマンドの前にReset_Requestコマンドをホストアプリケーションに送信する必要があります。

ホストアプリケーションがAnybus CompactCom 40 EtherCATをリセットする必要がある場合、リセット処理は下のフローチャートに示すように、ホストアプリケーションで実装する必要があります。

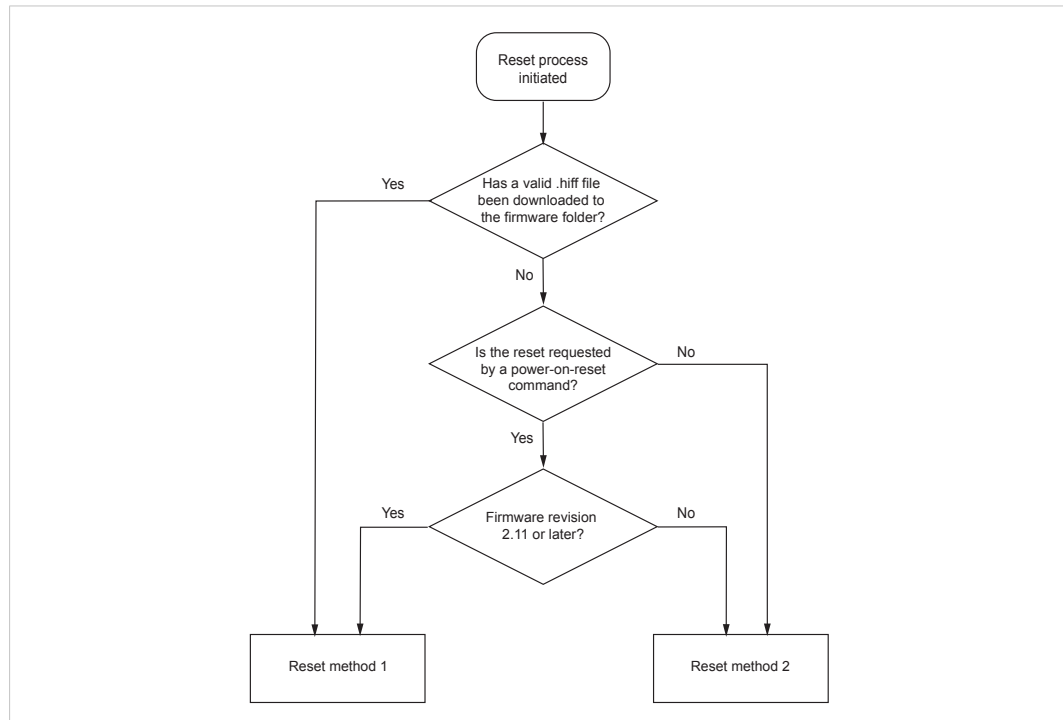


図 2

リセット方法1： リセット信号が有効になり、Anybus CompactComの初期化が可能になるまで有効の状態が保たれます。

リセット方法2： 10 μsより長い短パルスが、リセット信号に適用されます。その後少なくとも20 ms後にリセット信号が再び有効になり、Anybus CompactComの初期化が可能になるまで有効の状態が保たれます。

3.7.2 メーカーのパラメータをデフォルトに戻す

本モジュールは、「Restore Manufacturer Parameters to Default」要求をネットワークから受け取ると、CmdExt[1]設定01h (工場出荷時状態へのリセット) のリセットコマンドをアプリケーションオブジェクト (FFh) に発行します。

工場出荷時状態へのリセットを実行できるのは、EtherCATがPREOPERATIONALの状態にあるときだけです。PREOPERATIONAL以外の状態のときにリセットを実行すると、SDOアボートコード08000020h (無効な状態) が生成されます。

下記も参照してください。

- 標準オブジェクト, ページ 23、エントリ 1011h (パラメータの復元)

3.8 設定ステーションエイリアス (ノードアドレス)

設定ステーションエイリアス (ノードアドレス) の範囲は1～65535です。アドレス0は、その機器がまだ設定されていないことを表します。設定ステーションエイリアスは、スレーブのEEPROMに格納されます。このエイリアスは、マスターによりノードアドレスとして使用されます。

ほとんどのアプリケーションでは、設定ステーションエイリアスを変更しないことが推奨されます。ただし、ネットワークから各スレーブにアドレスを割り当てることは可能です。

3.9 Device ID

Device IDは、マスターがスレーブを明示的に識別するのに使用されます。このDevice IDは、例えばシステム稼働中に故障した機器を交換する場合 (いわゆるHotConnect用途) に役立ちます。あらかじめ設定しておいた機器をネットワークに投入可能です。また、新たに投入した機器のIDを故障した機器に割り当てられていたIDに設定します。

また、同じ種類の機器がネットワーク上に複数存在する場合にケーブルの挿し間違いを防ぐのにも役立ちます。

Device IDの範囲は1～65535です。アドレス0は、その機器がまだ設定されていないことを表します。この値は、ネットワーク設定オブジェクトのインスタンス1を使用して設定できます。



Anybus CompactCom M30 EtherCATでは、ネットワークコンフィグレーションオブジェクトのインスタンス3がDevice IDに使用されていました。

下記も参照してください。

- [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 40](#)

3.10 モジュラーデバイスプロファイル

Anybus CompactCom 40 EtherCATはモジュラーデバイスプロファイルに対応しています。これはモジュラーデバイスオブジェクトがアプリケーションに実装されている場合に有効になります。このプロファイルを実行すると、モジュールはスロット0のカプラーも含め最大63のスロットをサポートします。ネットワークからアクセス可能な最大ADI数は16383です。

デバイスタイプオブジェクト (1000h) の値は00005001hに変更されます。

モジュラーデバイスプロファイルを有効にすると、アセンブリマッピングオブジェクトの設定が無効になります (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) 。

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 (『Anybus CompactCom 40 Software Design Guide』)
- [モジュラーデバイスプロファイルのオブジェクトエントリ, ページ 31](#)

4 オブジェクトディクショナリ (CANopen over EtherCAT)

4.1 標準オブジェクト

4.1.1 概要

DS301通信プロファイルに従って、標準オブジェクトディクショナリが実装されています。なお、一部のオブジェクトエントリは、EtherCATオブジェクト (F5h) および診断オブジェクト (02h) の設定に対応しています。

4.1.2 オブジェクトエントリ

インデックス	オブジェクト名	サブインデックス	説明	タイプ	アクセス	注意事項
1000h	Device Type	00h	デバイスタイプ	U32	RO	デフォルトは0000 0000h (プロファイルなし)。必要に応じてホストアプリケーションに実装できるEtherCATオブジェクトを通じて管理できます。 EtherCATオブジェクト (F5h) 、 ページ 81 を参照してください。 ホストアプリケーションにモジュラーデバイスオブジェクトが実装されている場合、デフォルト値は0000 5001hになります。
1001h	Error register	00h	エラーレジスタ	U8	RO	この情報は診断オブジェクトで管理します。 診断オブジェクト (02h) 、 ページ 36 を参照。
1003h	Pre-defined error field	00h	エラー数	U8	RW	
		01h ~ 05h	エラーフィールド	U32	RO	
1008h	Manufacturer device name	00h	メーカーのデバイス名	可視文字列	RO	これらのエントリはEtherCATオブジェクトで管理されます。EtherCATオブジェクトはオプションでホストアプリケーションに実装できます。 EtherCATオブジェクト (F5h) 、 ページ 81 を参照してください。
1009h	Manufacturer hardware version	00h	メーカーのハードウェアバージョン	可視文字列	RO	
100Ah	Manufacturer software version	00h	メーカーのソフトウェアバージョン	可視文字列	RO	
1011h	Restore parameters	00h	最大のサブインデックスをサポート	U8	RO	01h
		01h	すべてのデフォルトパラメータを復元	U32	RW	-
1018h	Identity object	00h	エントリ数	U8	RO	エントリ数
		01h	ベンダーID	U32	RO	これらのエントリはEtherCATオブジェクトで管理されます。EtherCATオブジェクトはオプションでホストアプリケーションに実装できます。 EtherCATオブジェクト (F5h) 、 ページ 81 を参照してください。
		02h	製品コード	U32	RO	
		03h	リビジョン番号	U32	RO	
		04h	シリアル番号	U32	RO	
10F1h	Error Settings object	00h	エントリ数	U8	RO	
		02h	同期エラーカウンター上限	U32	RW	

インデックス	オブジェクト名	サブインデックス	説明	タイプ	アクセス	注意事項
1600h - 1xxxh	Receive PDO mapping	00h	PDOにマッピングされたアプリケーションオブジェクトの数	U8	RO/RW	マッピングされたオブジェクトの数 (0 ~ 254) 。詳細は PDOに対するADIのマッピング, ページ 26 を参照。 Receive PDO mappingは、動的プロセスデータがアプリケーションによってサポートされている場合に書き込むことができます (再マッピングコマンド) 。注 : PREOPデバイス状態でのみ書き込み可能。
		01h	マッピングされたオブジェクト #1	U32	RO/RW	-
		02h	マッピングされたオブジェクト #2	U32	RO/RW	-
		-
		NNh	マッピングされたオブジェクト #NN	U32	RO/RW	-
1A00h - 1xxxh	Transmit PDO mapping	00h	PDOにマッピングされたアプリケーションオブジェクトの数	U8	RO/RW	マッピングされたオブジェクトの数 (0 ~ 254) 。詳細は PDOに対するADIのマッピング, ページ 26 を参照。 Transmit PDO mappingは、動的プロセスデータがアプリケーションによってサポートされている場合に書き込むことができます (再マッピングコマンド) 。注 : PREOPデバイス状態でのみ書き込み可能。
		01h	マッピングされたオブジェクト #1	U32	RO/RW	-
		02h	マッピングされたオブジェクト #2	U32	RO/RW	-
		-
		NNh	マッピングされたオブジェクト #NN	U32	RO/RW	-
1C00h	Sync Manager Communication Type	00h	エントリ数	U8	RO	4
		01h	メールボックスの書き込み	U8	RO	1
		02h	メールボックスの読み込み	U8	RO	2
		03h	プロセスデータの出力	U8	RO	3
		04h	プロセスデータの入力	U8	RO	4
1C12h	Sync Manager Rx PDO Assign	00h	割り当てられたPDOの数	U8	RO/RW	静的なPDOマッピングを使用している場合、このサブインデックスはリードオンリーです。動的PDOマッピングを使用する場合は、書き込み可能です (PREOPデバイス状態でのみ書き込み可能) 。
		01h - NNh	割り当てられたPDO	U16	RO/RW	
1C13h	Sync Manager Tx PDO Assign	00h	割り当てられたPDOの数	U8	RO/RW	複数の同期モードがサポートされている場合、このエントリは書き込み可能です (PREOPデバイス状態でのみ書き込み可能) 。
		01h - NNh	割り当てられたPDO	U16	RO/RW	

インデックス	オブジェクト名	サブインデックス	説明	タイプ	アクセス	注意事項
1C32h	SM output parameter	00h	サポートされている最大のサブインデックス	U8	RO	12 (0Bh)
		01h	Sync mode	U16	RO/RW	00h: Free Run 02h: DC Sync0 SYNCオブジェクト (EEh) , ページ 79 を参照してください。
		02h	Cycle time	U32	RW	サイクルタイム (単位 : ナノ秒)
		03h	シフトタイム	U32	RW	シフトタイム (単位 : ナノ秒)
		04h	サポートされている同期の種類	U16	RO	ビット0がセット: FREE_RUNに対応 ビット2がセット: DC Sync0に対応。 ビット5がセット: ローカルタイマーによる出力シフト 他のビットはすべて0にセットされます。 SYNCオブジェクト (EEh) , ページ 79 を参照してください。
		05h	最小サイクルタイム	U32	RO	最小サイクルタイム (単位 : ナノ秒)。
		06h	出力計算およびコピー時間	U32	RO	出力計算およびコピー時間 (単位 : ナノ秒)。
		09h	遅延時間	U32	RO	遅延時間 (単位 : ナノ秒)。常に0に設定されます。
		0Ch	サイクルタイムが短すぎる	U16	RO	サイクルタイムが短すぎる
1C33h	SM input parameter	00h	サポートされている最大のサブインデックス	U8	RO	12 (0Bh)
		01h	Sync mode	U16	RO/RW	00h: Free Run 02h: DC Sync0 SYNCオブジェクト (EEh) , ページ 79 を参照してください。
		02h	Cycle time	U32	RW	サイクルタイム (単位 : ナノ秒)。1C32h、サブインデックス2と同じ値。
		03h	シフトタイム	U32	RW	シフトタイム (単位 : ナノ秒)。
		04h	サポートされている同期の種類	U16	RO	ビット0がセット: FREE_RUNに対応 ビット2がセット: DC Sync0に対応。 ビット5がセット: ローカルタイマーによる入力シフト 他のビットはすべて0にセットされます。 SYNCオブジェクト (EEh) , ページ 79 を参照してください。
		05h	最小サイクルタイム	U32	RO	最小サイクルタイム (単位 : ナノ秒)。1C32h、サブインデックス5と同じ値。
		06h	入力計算およびコピー時間	U32	RO	入力計算およびコピー時間 (単位 : ナノ秒)。
		0Ch	サイクルタイムが短すぎる	U16	RO	サイクルタイムが短すぎる。1C32h、サブインデックス12 (0Bh) と同じ値。

PDOに対するADIのマッピング

Receive PDO mappingオブジェクト (1600h ~ 1xxxh) と Transmit PDO mappingオブジェクト (1A00h ~ 1xxxh) は、ホストアプリケーションのセットアップ方法に応じて以下のように設定されます。

Mode	アクセス	オブジェクトの数 (各方向)	オブジェクトあたりのサブインデックスの数	注意事項
一般、静的マッピング	RO	1 ~ 6 セットアップ中にアプリケーションによってマッピングされるADIマッピングアイテムの数によって異なります。各PDOは254個のADIマッピングアイテムを保持できます。	1 ~ 254 セットアップ中にアプリケーションによってマッピングされるADIマッピングアイテムの数によって異なります。一度に1つのPDOマッピングオブジェクトにマッピングアイテムが格納されます。	
一般、動的マッピング	RW	1 ~ 6 セットアップ中にアプリケーションによってマッピングされるADIマッピングアイテムの数によって異なります。各PDOは254個のADIマッピングアイテムを保持できます。	254 (6番目のオブジェクトを除く。最大エントリ数は1486なので、6番目のオブジェクトのサブインデックスの数は216になる)	TwinCAT 3 ツール (または 2.11以降のバージョン) が使用されている場合、ツールの制限により、最大エントリ数は1473バイトとなります。
アセンブリマッピングオブジェクトがホストに実装されている	RO/RW	その方向のアセンブリマッピングインスタンスの数 (最大63)	1486/ (オブジェクトの数) (最大254)	詳細は アセンブリマッピングオブジェクト (EBh) 、 ページ 78 を参照してください。 アクセスは、対応するアセンブリインスタンスが静的な場合はRO、動的な場合はRW TwinCAT 3 ツール (または 2.11以降のバージョン) が使用されている場合、ツールの制限により、最大エントリ数は1473バイトとなります。
モジュラーデバイス、静的マッピング	RO	その方向でマッピング可能なオブジェクトを持つモジュールの数と同じ (最大63)	セットアップ中にその方向でマッピングされたADIの数と同じ	
モジュラーデバイス、動的マッピング	RW	その方向でマッピング可能なオブジェクトを持つモジュールの数と同じ (最大63)	1486/ (オブジェクトの数) (最大254)	TwinCAT 3 ツール (または 2.11以降のバージョン) が使用されている場合、ツールの制限により、最大エントリ数は1473バイトとなります。

一般モードおよびモジュラーデバイスプロファイルモードでは、ADIからPDOへのマッピングは起動時にアプリケーションによって実行されることに注意してください。また、アセンブリマッピングオブジェクトとモジュラーデバイスオブジェクトの両方がホストに実装されている場合は、モジュラーデバイスプロファイルモードが有効になり、アセンブリマッピングオブジェクトの設定は無効になることにも注意してください。

PDO Assignオブジェクト (1C12hと1C13h) は、ホストアプリケーションのセットアップ方法に応じて以下のように設定されます。

Mode	アクセス	オブジェクトあたりのサブインデックスの数	内容
一般、静的マッピング	RO	その方向のPDOマッピングオブジェクトの数と同じ。	その方向のすべてのPDOマッピングオブジェクト。
一般、動的マッピング	RW	その方向のPDOマッピングオブジェクトの数と同じ。	その方向のすべてのPDO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がホストに実装されている	RW	その方向のPDOマッピングオブジェクトの数と同じ。	その方向の最初のPDO。
モジュラーデバイス、静的マッピング	RO	その方向のPDOマッピングオブジェクトの数と同じ。	その方向のすべてのPDOマッピングオブジェクト。
モジュラーデバイス、動的マッピング	RW	その方向のPDOマッピングオブジェクトの数と同じ。	その方向のすべてのPDOマッピングオブジェクト。

4.2 メーカー及び、プロファイル固有オブジェクト

4.2.1 概要

メーカー固有の範囲 (2001h ~ FFFFh) にある各オブジェクトエントリは、アプリケーションデータオブジェクト (FEh) 内のインスタンス (ADI) に対応します。すなわち、これらのオブジェクトにネットワークからアクセスすると、ホストアプリケーションに対するオブジェクト要求が発生します。エラー発生時、ホストアプリケーションから返送されたエラーコードは対応するCANopenアボートコードに変換されます。



これらのオブジェクトエントリにアクセスすると、オブジェクトからホストアプリケーションへのアクセスが発生するため、SDOのタイムアウト値を計算する際は、ホストインターフェースの通信に要する時間を考慮する必要があります。

4.2.2 ネットワークデータのフォーマット

データは、以下のようにネットワーク固有の形式からAnybusのデータ形式に変換されます。

Anybusデータ型	ネットワークデータタイプ	
	Number of sub elements = 1	Number of sub elements > 1
BOOL	UNSIGNED8	OCTET_STRING
SINT8	INTEGER8	ARRAY_OF_SINT
SINT16	INTEGER16	ARRAY_OF_INT
SINT32	INTEGER32	ARRAY_OF_DINT
UINT8	UNSIGNED8	OCTET_STRING
UINT16	UNSIGNED16	ARRAY_OF_UINT
UINT32	UNSIGNED32	ARRAY_OF_UDINT
CHAR	VISIBLE_STRING	VISIBLE_STRING
ENUM	UNSIGNED8またはENUM	OCTET_STRING
BITS8	BITARR8	OCTET_STRING
BITS16	BITARR16	ARRAY_OF_UINT
BITS32	BITARR32	ARRAY_OF_UDINT
OCTET	OCTET_STRING	OCTET_STRING
SINT64	INTEGER64	OCTET_STRING
UINT64	UNSIGNED64	OCTET_STRING
FLOAT	REAL32	OCTET_STRING
PADO-16	NULL	非対応
BOOL1	BOOL	非対応
BIT1 - BIT7	BIT1 - BIT7	非対応

複数の要素を持つADIは、配列 (すべての要素が同じデータ型を共有) またはレコード (各要素が異なるデータ型になる) として表されます。ただし、常にVISIBLE_STRINGとして表されるCHARと常にOCTET_STRINGとして表されるOCTETは例外です。

要素を1つだけ持つADIは単変数で表されます。ただし、常にVISIBLE_STRINGとして表されるCHARと常にOCTET_STRINGとして表されるOCTETは例外です。

4.2.3 エラーコード

アプリケーションのオブジェクトがモジュールから要求されたときにエラーが発生した場合、返されたエラーコードは以下のようにCANopenアポートコードに変換されます。

Anybus CompactComエラーコード	CANopenアポートコード	説明 (CANopen)
予約	N/A	-アプリケーションのオブジェクトがモジュールから要求されたときにエラーが発生した場合、返されたエラーコードは以下のようにCANopenアポートコードに変換されます。
Fragmentation error (serial mode)	N/A	-
Invalid message format	N/A	-
Unsupported object	0602 0000h	オブジェクトがオブジェクトディクショナリに存在しません。
Unsupported instance	0602 0000h	オブジェクトがオブジェクトディクショナリに存在しません。
Unsupported command	0604 0043h	一般的なパラメータ不適合の理由。
Invalid CmdExt[0]	0602 0000h	オブジェクトがオブジェクトディクショナリに存在しません。 (ADIアクセス)
Invalid CmdExt[1]	0609 0011h	サブインデックスが存在しません。 (ADIアクセス)
Attribute not settable	0601 0002h	リードオンリーオブジェクトに書き込もうとしました。
Attribute not gettable	0601 0001h	ライトオンリーオブジェクトを読み出そうとしました。
Too much data	0607 0012h	データ型が一致しません。サービスパラメータの長さが長すぎます。
Not enough data	0607 0013h	データ型が一致しません。サービスパラメータの長さが短すぎます。
Out of range	0609 0030h	パラメータの値範囲を超えています (ライトアクセスの場合のみ)。
Invalid state	0800 0022h	現在の機器の状態が理由で、データをアプリケーションに転送または保存できません。
Out of resources	0504 0005h	メモリ不足
Value too high	0609 0031h	パラメータの値が上限を超えています (ライトアクセスの場合のみ)。
Value too low	0609 0032h	パラメータの値が下限を下回っています (ライトアクセスの場合のみ)。
Write access to a read process data mapped ADI	0601 0006h	RxPDOにマッピングされたオブジェクトのSDOダウンロードがブロックされました。
Protected access	0800 0021h	ローカル制御のため、データの読み取りまたは格納ができません
Object Specific Error	0800 0000h	General error

対応するエラーコードがCANopenで特定できない場合、デフォルトのエラーコードは一般エラー (0800 000h) となります。

4.2.4 オブジェクトエントリ

ADIの厳密な表現は、その要素数によって異なります。以下の例では、ADI 0002hと0004hはそれぞれ要素を1つだけ持つため、これらのADIは配列ではなく単変数で表されます。その他のADIは (同じデータ型の) 複数の要素を持つため、これらのADIは配列で表されます。ADIが複数の要素を持ち、それらのデータ型が異なる場合は、レコードで表されます。

インデックス	オブジェクト名	サブインデックス	説明	タイプ	アクセス
2001h	ADI 0001h	00h	エントリ数 (NNh)	U8	RO
		01h	ADI値 (アトリビュート#5)	-	-
		02h	複数の要素を持つADI (配列) は、複数のサブインデックスで表されます。		
		...	ADI値のデータタイプとアクセス権は、ADI自身が決定します。		
		NNh			
2002h	ADI 0002h	00h	ADI値 (アトリビュート#5)	-	-
2003h	ADI 0003h	00h	エントリ数 (NNh)	U8	RO
		01h	ADI値 (アトリビュート#5)	-	-
		02h	複数の要素を持つADI (配列) は、複数のサブインデックスで表されます。		
		...			
		NNh			
2004h	ADI 0004h	00h	ADI値 (アトリビュート#5)	-	-
2005h	ADI 0005h	00h	エントリ数 (NNh)	U8	RO
		01h	ADI値 (アトリビュート#5)	-	-
		02h	複数の要素を持つADI (配列) は、複数のサブインデックスで表されます。		
		...			
		NNh			
...
5FFFh	ADI 3FFFh	00h	エントリ数 (NNh)	U8	RO
		01h	ADI値 (アトリビュート#5)	-	-
		02h	複数の要素を持つADI (配列) は、複数のサブインデックスで表されます。		
		...			
		NNh			

4.2.5 Fail Safe over EtherCAT、オブジェクトエントリ

Fail Safe over EtherCATが有効になっている場合は、以下のオブジェクトが実装されます。

インデックス	オブジェクト名	サブインデックス	説明	タイプ	アクセス	注意事項
F980h	Device safety address	00h	Device safety address	UNSIGNED8	RO	-

4.2.6 モジュラーデバイスプロファイルのオブジェクトエントリ

次の表に示すオブジェクトは、モジュラーデバイスプロファイルモードが有効な場合に実装されます。

インデックス	オブジェクト名	サブインデックス	説明	タイプ	アクセス	注意事項
6000h - 6FFFh	Input data	任意	カブラーを除くすべてのモジュールの、ライトプロセスデータにマッピング可能なADIは、この範囲で表されます。	任意	R, RW	詳細については、 ADIからSDOへの変換 、 ページ 32 を参照してください。
7000h- 7FFFh	Output data	任意	カブラーを除くすべてのモジュールの、リードプロセスデータにマッピング可能なADIは、この範囲で表されます。	任意	W, RW	詳細については、 ADIからSDOへの変換 、 ページ 32 を参照してください。
9nnnh	Information data	任意	情報オブジェクト。スロットを占めるモジュールごとに1つずつ存在します (カブラーは除く)。	任意	RW	詳細については、 モジュール識別オブジェクト 、 ページ 33 を参照。
F000h	モジュラーデバイスプロファイル	00h	エントリ数 (NNh)	U8	R	値: 5
		01h	インデックス距離	U16	R	この値は、各スロットに割り当てられているオブジェクトの数を決定します。この値はすべてのモジュールで同じです。したがって、これは2つのスロット間のインデックス距離になります。 値: "Number of ADIs per slot"、モジュラーデバイスオブジェクトのアトリビュート #12。詳細については、『Anybus CompactCom 40 Software Design Guide』を参照。
		02h	モジュールの最大数	U16	R	値: "Number of slots"、モジュラーデバイスオブジェクトのアトリビュート11。詳細については、『Anybus CompactCom 40 Software Design Guide』を参照。
		04h	基本情報	U32	R	値: 0000 0700h (サブインデックス9、10、11は、9nnnhモジュール識別オブジェクトでサポートされています)
		05h	デバイスのモジュールPDOグループ	U16	R	0に設定すると、カブラーのプロセスデータが強制的に他のプロセスデータの前に配置されます。これにより、モジュラーデバイスホストオブジェクトに対する統合が向上します。
F030h	Configured Module Ident List	00h	エントリの数 (スロットの数-1)	U8	R	マスターはこれらのオブジェクトに設定モジュールリストを書き込みます。そのため、スレーブは期待されるモジュール設定を実際の設定と比較できます。
		01h	ポジション1 (スロット1) に設定されたモジュールのモジュールID。	U32	RW	
				
		0nh	ポジションn (スロットn) に設定されたモジュールのモジュールID。	U32	RW	

インデックス	オブジェクト名	サブインデックス	説明	タイプ	アクセス	注意事項
F050h	Detected Module Ident List	00h	エントリの数 (スロットの数-1)	U8	R	このオブジェクトには、アプリケーションから検出された、スロットに配置されているモジュールの情報が格納されます。
		01h	ポジション1 (スロット1) に設定されたモジュールのモジュールID。	U32	RW	
				
		0nh	ポジションn (スロットn) に設定されたモジュールのモジュールID。	U32	RW	
F600h - F6FFh	カプラーの入力データ領域	任意	カプラーのライトプロセスにマッピング可能なADIは、この範囲で表されます。	任意	R, RW	-
F700h - F7FFh	カプラーの出力データ領域	任意	カプラーのリードプロセスにマッピング可能なADIは、この範囲で表されます。	任意	W, RW	-

Configured Module Ident List (F030h) が Detected Module Ident List (F050h) と一致しない場合、モジュールは ALStatusCode レジスタを 0070h に設定することによって設定の不一致を通知します。モジュールは SAFE-OPERATIONAL 状態にはなりません。



このリスト比較は、EtherCAT オブジェクトの「ID リストの比較」アトリビュート (アトリビュート 22) を FALSE に設定することでスキップできます。

ADI から SDO への変換

モジュラーデバイスプロファイルでは、すべての ADI を番号順にマッピングする必要があります。1 つのスロットにマッピングされる ADI の数はモジュラーデバイスオブジェクトで定義されており、同じ数のオブジェクトが各スロットに割り当てられます。ADI はリードとライトのどちらにもマッピング可能かに応じて、異なるオブジェクト範囲にマッピングされます。読み出しと書き込みのどちらにもマッピング可能な ADI は、両方の範囲にマッピングされます。SDO は番号順に割り当てられますが、占める範囲は種類に応じて異なることに注意してください。

リードとライトのどちらにもマッピングできない ADI は SDO にマッピングされません。その結果、次の表に示すように「空の SDO」になります。

モジュール	ADI	タイプ	SDO
0 (カプラー)	1	ライトマッピング可能	F600h
	2	リードマッピング可能	F701h
	3	ライトマッピング可能	F602h
	4	リードマッピング可能	F703h
	5	マッピング不可	-
1	6	リードマッピング可能	7000h
	7	ライトマッピング可能	6001h
	8	書き込み可能	-
	9	リードオンリー	-
	10	リードマッピング可能	7004h
2	11	書き込み可能	-
	12	リードオンリー	-
	-	-	-
	14	ライトマッピング可能	6008h
	15	リードおよびライトマッピング可能	6009h および 7009h

モジュール識別オブジェクト

各モジュールの9nnnh範囲にある最初のSDOIは、次の表に従ってあらかじめ定義されています。

サブインデックス	タイプ	アクセス	名前および説明
00h (0)	U8	R	サポートされている最大サブインデックス。 値: 11 (0Bh)
09h (9)	U16	R	モジュールPDOグループ。 値: 1. (PDOグループはカプラーを除くすべてのモジュールで1に設定されます。これにより、カプラーのデータをモジュールのデータより前に置くことができます。)
0Ah (10)	U32	R	モジュールID (ホストアプリケーションに従ったモジュールのモジュールID)
0Bh (11)	U16	r	スロット (モジュール番号)

PDOマッピング

Receive PDO mappingオブジェクトとTransmit PDO mappingオブジェクトは、ホストアプリケーションのセットアップ方法に応じて設定されます。16xxhシリーズの1つのオブジェクトは各モジュールに対して作成され、少なくとも1つのリードマッピング可能なADIを保持します。オブジェクトの番号は1600h + スロット番号- 1になります。1Axxhシリーズの1つのオブジェクトは各モジュールに対して作成され、少なくとも1つのライトマッピング可能なADIを保持します。オブジェクトの番号は1A00h + スロット番号- 1になります。

カプラーがリードまたはライトマッピング可能なADIを保持する場合、それらを格納するオブジェクトが作成されます。カプラー用のオブジェクトは、他のすべてのマッピングオブジェクトが作成された後に作成されます。

詳細については、[PDOに対するADIのマッピング](#), ページ 26を参照してください。

5 Anybus モジュールオブジェクト

5.1 基本情報

この章では、本モジュールで実装されているAnybusモジュールオブジェクトについて説明します。

標準オブジェクト:

- [Anybusオブジェクト \(01h \)](#) , ページ 35
- [診断オブジェクト \(02h \)](#) , ページ 36
- [ネットワーク オブジェクト \(03h \)](#) , ページ 38
- [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 40
- [ソケットインターフェースオブジェクト \(07h \)](#) , ページ 45
- [SMTPクライアントオブジェクト \(09h \)](#) , ページ 62
- [ファイルシステムインターフェースオブジェクト \(0Ah \)](#) 、 『Anybus CompactCom 40 Software Design Guide』 を参照
- [ネットワークイーサネットオブジェクト \(0Ch \)](#) , ページ 67
- [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 69

ネットワーク固有のオブジェクト:

(なし)

5.2 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共通するすべてのAnybusデータをまとめています。そして、全般的な『Anybus CompactCom 40 Software Design Guide』に詳細を記載しています。

サポートコマンド

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

このオブジェクトは共通するすべてのAnybusデータをまとめています。そして、全般的な『Anybus CompactCom 40 Software Design Guide』に詳細を記載しています。

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 (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』を参照してください。
17	Virtual attributes	Get/Set		
18	Black list/White list	Get/Set		
19	Network time	Get	UINT64	ナノ秒で表した64ビット値。 基準: 2000年1月1日、12 : 00 AM Network timeアトリビュートには、EtherCATスレーブコントローラーのDCシステム時刻レジスタの値が格納されます。

5.3 診断オブジェクト (02h)

カテゴリ

拡張

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標準的な手段を提供します。このオブジェクトについては、『Anybus CompactCom 40 Software Design Guide』で詳しく説明されています。

EMCYオブジェクト (緊急オブジェクト) は、診断インスタンスが作成または削除されるたびにネットワーク上で送信されます。

サポートコマンド

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	タイプ	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 (復旧不可能な重大なイベントのために1つのインスタンスが予約されています)
12	Supported functionality	Get	BITS32	ビット0: 0 (モジュールはイベントのラッチをサポートしていません) ビット1 ~ 31: 0

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	Severity	Get	UINT8	『Anybus CompactCom 40 Software Design Guide』を参照。
2	Event Code	Get	UINT8	
3	NW specific extension	Get	UINT8の配列	CANopen固有のEMCYコード (2バイト)
4 - 7	(未使用)			

インスタンスを作成すると (すなわち、診断イベントが発生すると)、以下の処理が実行されます。

- 2つの方法のどちらかで、オブジェクトエントリ1003h (定義済みのエラーフィールド) に、新たなエントリが作成されます。

- Event Codeが00h — FEhの場合 :

MSB	(UINT32)	LSB
(未使用)	(未使用)	Event Code
		00h

- Event CodeがFFh (ネットワーク固有) の場合 :

MSB	(UINT32)	LSB
(未使用)	(未使用)	エラーコード (下位バイト)

2. エラーレジスタ (オブジェクトエントリ1001h) のビット情報が設定されます。

ビット	説明	ビットの設定条件
0	一般エラー	このオブジェクト内の他のエラービットが設定されたときに常に設定されます。
1	電流	イベントコードが ⁶ 20h～23hである。 または イベントコードが ⁶ FFhで、なおかつNW固有情報の上位バイトが ⁶ 20h～23hである。
2	電圧	イベントコードが ⁶ 30h～33hである。 または イベントコードが ⁶ FFhで、なおかつNW固有情報の上位バイトが ⁶ 30h～33hである。
3	温度	イベントコードが ⁶ 40h～42hである。 または イベントコードが ⁶ FFhで、なおかつNW固有情報の上位バイトが ⁶ 40h～42hである。
4	通信エラー	イベントコードが ⁶ 80h～82hである。 または イベントコードが ⁶ FFhで、なおかつNW固有情報の上位バイトが ⁶ 80h～82hである。 または Anybusの状態が ⁶ ERRORである。
5	デバイスプロファイル固有	常に0
6	予約	常に0
7	メーカー固有	イベントコードが ⁶ FFhである。 かつ NW固有情報の上位バイトが ⁶ FFhである。

3. 診断インスタンスがWAIT_PROCESS以上の状態で作成された場合、EMCYオブジェクトが次の情報とともにネットワークに送信されます。

バイト0	バイト1	バイト2	バイト3	バイト4	バイト5	バイト6	バイト7
00h	Event Code	エラーレジスタ (1001h)	メーカー固有フィールド (未使用)				

SETUPまたはNW_INITのいずれかの状態でインスタンスが作成された場合は、EMCYオブジェクトは送信されません。

復旧不可能な重大なイベントを作成すると、Anybusモジュールが⁶EXCEPTION状態になるため、EMCYメッセージはバス上に送信されません。

アトリビュート3が実装されている場合、バイト0および1 (00h+イベントコード) はアトリビュート3の値に置き換えられます。

診断インスタンスが削除されると、エラーコードが⁶0000h (「エラーリセット」) のEMCYオブジェクトが送信されます。

5.4 ネットワーク オブジェクト (03h)

カテゴリ

基本

オブジェクトの説明

このオブジェクトについての詳細は、全般的な『Anybus CompactCom 40 Software Design Guide』をご参照ください。

サポートコマンド

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	0087h
2	Network type string	Get	CHAR配列	"EtherCAT"
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 (単位 : バイト) Map_ADI_Write_Area、Map_ADI_Write_Ext_Area、およびRemap_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『Anybus CompactCom 40 Software Design Guide』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 (単位 : バイト) Map_ADI_Read_Area、Map_ADI_Read_Ext_Area、およびRemap_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『Anybus CompactCom 40 Software Design Guide』を参照してください。
7	Exception Information	Get	UINT8	本モジュールがEXCEPTION状態になったとき、追加情報がここに設定されます。下表に示す例外情報を参照してください。 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。
8... 10	(予約)	-		

Exception Information

値	説明
00h	(追加情報なし)
01h	(予約)
02h	
03h	
04h	
05h	
06h	アセンブリマッピングホストオブジェクトの実装が正しくありません (例 : アトリビュート11または12がサポートされていない)。
07h	アプリケーションはADI再マッピングコマンドをサポートしていますが、アプリケーションデータオブジェクトのオブジェクトアトリビュート11または12 ("No. of read process data mappable instances"または"No of write process data mappable instances") を要求したとき、またはアプリケーションデータオブジェクトに対してGet_Instance_Numbersコマンドを発行したときに、アプリケーションからエラー応答が返されました。
08h	ホストアプリケーションでのモジュラーデバイスオブジェクトの実装が正しくありません (例 : Get_Listコマンドの発行時にエラー応答が返された)。
09h	Anybus IPの実行時にMACアドレスが欠落しています。

5.5 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。

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。

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 88

サポートコマンド

オブジェクト :	Get_Attribute
	Reset
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	タイプ	値
1	Name	Get	CHAR配列	「Network configuration」
2	Revision	Get	UINT8	01h
3	Number of instances	Get	UINT16	12
4	Highest instance no.	Get	UINT16	15

インスタンスアトリビュート (インスタンス#1、Device ID)

拡張

[Device ID](#), ページ 22も参照してください。

変更は直ちに有効になります。

#	名前	アクセス	タイプ	値
1	Name	Get	CHAR配列	「Device ID」 Multilingual (多言語)。多言語文字列, ページ 44を参照。
2	Data type	Get	UINT8	05h (= UINT16)
3	Number of elements	Get	UINT8	01h (1要素)
4	Descriptor	Get	UINT8	03h (読み出し/書き込みアクセス)
5	値	Get/Set	UINT16	1 ~ 65535 : 有効なネットワークアドレス 0 : 機器は未設定 (デフォルト)
6	Configured Value	Get	UINT16	Device IDの設定値。この値は常にアトリビュート#5の値と等しくなります。

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配列	「IP address」 (Multilingual (多言語))。 多言語文字列, ページ 44 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	値の有効範囲: 0.0.0.0 ~ 255.255.255.255 (デフォルト =0.0.0.0)



このアトリビュートは認証の問題の原因となるため、電源を入れるたびにアプリケーションによって設定される
ことのないようにする必要があります。

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配列	「Subnet mask」 (Multilingual (多言語))。 多言語文字列, ページ 44 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	値の有効範囲: 0.0.0.0 ~ 255.255.255.255 (デフォルト =0.0.0.0)



このアトリビュートは認証の問題の原因となるため、電源を入れるたびにアプリケーションによって設定される
ことのないようにする必要があります。

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)

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配列	「Gateway」 (Multilingual (多言語))。 多言語文字列, ページ 44 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	値の有効範囲: 0.0.0.0 ~ 255.255.255.255 (デフォルト =0.0.0.0)



このアトリビュートは認証の問題の原因となるため、電源を入れるたびにアプリケーションによって設定される
ことのないようにする必要があります。

インスタンスアトリビュート (インスタンス#6、DHCP)

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配列	「DHCP」 (Multilingual (多言語)。 多言語文字列, ページ 44 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><tr><th>値</th><th>文字列</th><th>意味</th></tr><tr><td>00h</td><td>「Disable」</td><td>DHCP無効 (デフォルト)</td></tr><tr><td>01h</td><td>「Enable」</td><td>DHCP有効 (Multilingual (多言語)。 多言語文字列, ページ 44を参照)</td></tr></table>	値	文字列	意味	00h	「Disable」	DHCP無効 (デフォルト)	01h	「Enable」	DHCP有効 (Multilingual (多言語)。 多言語文字列, ページ 44 を参照)
値	文字列	意味											
00h	「Disable」	DHCP無効 (デフォルト)											
01h	「Enable」	DHCP有効 (Multilingual (多言語)。 多言語文字列, ページ 44 を参照)											

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適用されます。

#	名前	アクセス	データ型	説明
1	Name	Get	CHAR配列	「DNS1」 (Multilingual (多言語)。 多言語文字列, ページ 44 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	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適用されます。

#	名前	アクセス	データ型	説明
1	Name	Get	CHAR配列	「DNS2」 (Multilingual (多言語)。 多言語文字列, ページ 44 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	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適用されます。

#	名前	アクセス	データ型	説明
1	Name	Get	CHAR配列	「Host name」 (Multilingual (多言語)。 多言語文字列, ページ 44 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配列	ホスト名、64文字 (フルレングスになるようにスペースでパディング)

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適用されます。

#	名前	アクセス	データ型	説明
1	Name	Get	CHAR配列	「Host name」 (Multilingual (多言語)。 多言語文字列, ページ 44 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配列	ドメイン名、48文字 (フルレングスになるようにスペースでパディング)

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適用されます。

#	名前	アクセス	データ型	説明
1	Name	Get	CHAR配列	「SMTP server」 (Multilingual (多言語)。 多言語文字列, ページ 44 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配列	SMTPサーバアドレス、64文字。ドット付10進数フォーマットまたはサーバ名 (フルレングスになるようにスペースでパディング)

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アカウントユーザー名を格納します。変更はリセット後に適用されます。

#	名前	アクセス	データ型	説明
1	Name	Get	CHAR配列	「SMTP user」 (Multilingual (多言語))。多言語文字列, ページ 44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配列	SMTPアカウントユーザー名、64文字 (フルレングスになるようにスペースでパディング)

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アカウントパスワードを格納します。変更はリセット後に適用されます。

#	名前	アクセス	データ型	説明
1	Name	Get	CHAR配列	「SMTP Pswd」 (Multilingual (多言語))。多言語文字列, ページ 44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配列	ホスト名、64文字 (フルレングスになるようにスペースでパディング)

インスタンスアトリビュート (インスタンス#21、FSOEアドレス)

このインスタンスは、Fail Safe over EtherCAT実行時にFSOEアドレスを保持します。Valueアトリビュート (#5) に書き込まれるデータは、不揮発性メモリに保存されます。

#	名前	アクセス	データ型	説明
1	Name	Get	CHAR配列	FSOEアドレス (Multilingual (多言語))。多言語文字列, ページ 44を参照)
2	Data type	Get	UINT8	データ型 : UINT16
3	Number of elements	Get	UINT8	1つのデータ要素
4	Descriptor	Get	UINT8	ビット0: 1 = 読み取りアクセス ビット1: 1 = 書き込みアクセス
5	値	Get/Set	UINT16	ホストアプリケーションによって設定されるFSOEアドレス 範囲 : 1 ~ 65535

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
1	Device ID	Geräteadresse	ID Dispos.	ID Dispos.	ID appareil
21	FSOEアドレス	FSOE Adresse	Dirección FSoE	Indirizzo FSoE	FSoE Adresse

Reset

工場出荷時状態へのリセットコマンドをこのオブジェクトに発行すると、設定されたデバイスIDが0 (デフォルト値) にリセットされます。

5.6 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 60](#)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	値: Socket Type 00h SOCK_STREAM, NONBLOCKING (TCP) 01h SOCK_STREAM, BLOCKING (TCP) 02h SOCK_DGRAM, NONBLOCKING (UDP) 03h SOCK_DGRAM, BLOCKING (UDP)
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	State (TCPソケットのみ) : 値 ステート/説明 00h CLOSED 閉じた 01h LISTEN 接続待機中 02h SYN_SENT アクティブ、SYNを送受信済み 03h SYN_RECEIVED SYNを送受信済み 04h ESTABLISHED 確立済み。 05h CLOSE_WAIT FINを受信して、クローズ待ち 06h FIN_WAIT_1 クローズして、FINを送信済み 07h CLOSING 交換済みFINを閉じて、FIN ACK待ち 08h LAST_ACK FINを送信、クローズして、FIN ACK待ち 09h FIN_WAIT_2 クローズして、FINが確認された Ah TIME_WAIT クローズ後静かに待つ
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	ソケットはローカルアドレスを再使用可能 値 意味 1 有効 0 無効 (デフォルト)
9	Keep alive	Get/Set	BOOL	プロトコルがアイドルコネクションを探索します (TCPソケットのみ)。 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。 値 意味 1 有効 0 無効 (デフォルト)
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	IPマルチキャストループバック (UDPソケットのみ) 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 値 意味 1 有効 (デフォルト) 0 無効
12	(予約)			
13	TCP No Delay	Get/Set	BOOL	合体パケットへの送出を遅延させない (TCP)。 値 意味 1 遅延 (デフォルト) 0 遅延しない (ソケット上のNagleアルゴリズムを停止しない)
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]	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コネクションを優雅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) に設定します。
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: 受信チャネルのシャットダウン 送信チャネルのシャットダウン 送受信チャネルともにシャットダウン</td></tr></table>	値: 00h 01h 02h	モード: 受信チャネルのシャットダウン 送信チャネルのシャットダウン 送受信チャネルともにシャットダウン
値: 00h 01h 02h	モード: 受信チャネルのシャットダウン 送信チャネルのシャットダウン 送受信チャネルともにシャットダウン		

- 応答詳細

(データなし)

TCPコネクションを優雅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:

- 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。受信チャネルがまだ稼働していることに注意してください。
- 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。ホストが受信チャネルを終了したことを表します。ホストが受信チャ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。
- 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:

- 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ネルをクローズしたことを表します。
- 未送信データをホストに送るよう試みてください。
- 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、送信チャネルをシャットダウンします。
- 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 60](#)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) バイトで応答した場合、これはホストがコネクションを閉じたことを意味します。しかし送信チャネルはまだ有効であり、**Shutdown**と**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 60 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 60](#)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 60 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 60](#)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 60 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 60](#)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 60 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 60](#)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 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 60](#)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 60 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 60](#)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 60](#)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 60 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「All hosts group」を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	ENOBUFS	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	(reserved)	-
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	(reserved)	-
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を置換する場合が挙げられます。ソケットメッセージの最大サイズは1472です。255 バイトを超える場合は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。

分割プロトコルはメッセージレイヤーに実装され、シリアルホストインターフェースで使用す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、全般的な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とき、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

応答が分割されるとき、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

5.7 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配列	例 「someone@somewhere.com」
2	To	Get/Set	CHAR配列	例 「 someone.else@anywhere.net」
3	Subject	Get/Set	CHAR配列	例 「Important notice」
4	Message	Get/Set	CHAR配列	例 「Shut down the system」

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

Se [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スキャンエラー
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)

5.8 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Ethernet固有情報をアプリケーションに提供します。

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

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 88も参照)
2 ~ 3	(予約)			
4	MAC Address	Get	UINT8の配列	Device MACアドレス
5 ~ 6	(予約)			

インスタンスアトリビュート (インスタンス#2)

#	名前	アクセス	データ型	説明
1 ~ 4	(予約)			
5	Interface Counters	Get	UINT32 の配列	配列インデックスについては、下表を参照してください。
6	Media Counters	Get	UINT32 の配列	配列インデックスについては、下表を参照してください。

インスタンスアトリビュート (インスタンス#3)

(予約)

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チェックに失格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;	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

5.9 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識別子。 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」は以下のように表されます。最初のバイト = 0x02、2番目のバイト = 0x12、3番目のバイト = 0x03。
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。

#	名前	アクセス	データ型	説明
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り破棄されたAnybus CompactComモジュールからの (予期せぬ) 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オブジェクト内の例外コードが「Safety communication error」 (09h) に設定されている場合、詳細な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です。 フォーマット：バージョン「2.12」は以下のように表されます。 最初のバイト = 0x02、2番目のバイト = 0x0C

Exception Information

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

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HMSI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 EtherCAT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 EtherCAT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) 。	-

6 ホストアプリケーションオブジェクト

6.1 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。EtherCATの機能を拡張する為に、以下に示すオブジェクトがホストアプリケーションファームウェアにオブションで実装される事があります。

標準オブジェクト:

- アプリケーションオブジェクト (『Anybus CompactCom 40 Software Design Guide』を参照)
- 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 (『Anybus CompactCom 40 Software Design Guide』を参照)
- ファンクショナルセーフティオブジェクト ([E8h](#)), ページ 76
- アセンブリマッピングオブジェクト ([EBh](#)), ページ 78
- SYNCオブジェクト ([EEh](#)), ページ 79
- モジュラーデバイスオブジェクト (『Anybus CompactCom 40 Software Design Guide』を参照)
- 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)
- [Ethernet](#)ホストオブジェクト ([F9h](#)), ページ 88

ネットワーク固有のオブジェクト:

- [EtherCAT](#)オブジェクト ([F5h](#)), ページ 81

6.2 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）。</p> <p>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。</p> <p>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。</p> <p>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 CompactComIは上記の最小値に従い周期時間の妥当性を確認します。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 CompactComI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>

6.3 アセンブリマッピングオブジェクト (EBh)

カテゴリ

拡張

オブジェクトの説明

アプリケーションにこのオブジェクトが実装されている場合、アプリケーションの起動時に作成されたPDOマッピングがこのオブジェクトによって置き換えられます。元のマッピングの置き換えはPRE-OPERATIONAL状態からSAFE-OPERATIONAL状態に移移するときに行われます。このオブジェクトを実装するアプリケーションは、Remap_ADIコマンドをサポート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内の各インスタンスは1つのPDOに対応します。最初のリードアセンブリはオブジェクトディクショナリ内のオブジェクト1600hにマッピングされ、2番目は1601hにマッピングされて、以下同様に続きます。同様に、最初のライトアセンブリはオブジェクトディクショナリ内のオブジェクト1A00hにマッピングされ、2番目は1A01hにマッピングされて、以下同様に続きます。リードアセンブリとライトアセンブリのそれぞれについて最大64個のインスタンスがサポートされています。

次の表に、アセンブリマッピングオブジェクトの各種インスタンスに対するPDOマッピングオブジェクト番号の割り当て例を示します。

インスタンス番号	Direction	PDOマッピングオブジェクト番号
1	書き込み	1A00h
2	読み出し	1600h
3	書き込み	1A01h
4	読み出し	1601h
5	読み出し	1602h

アセンブリマッピングの各インスタンスは最大254個のADI要素をサポートし、EtherCAT上の1つの完全なPDOに対応します。

モジュラーデバイスオブジェクトがホストアプリケーションに実装されている場合（すなわち、モジュラーデバイスプロファイルが有効になっている場合）、このオブジェクトの設定は無視されます。

以下も参照してください。

- Anybus CompactCom 40 Software Design Guide、「Assembly Mapping Object」
PDOマッピングのアセンブリに関する[標準オブジェクト](#), ページ 23。

6.4 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、ホストアプリケーションのSYNCに関する設定を実装します。

このオブジェクトの実装は任意です。このオブジェクトが実装されていない場合、モジュールはEtherCATフリーランモードのみをサポートします。

同期機能のコンフィグレーション全体に関する問題がある場合、アプリケーションはアプリケーションステータスレジスタでその旨を通知する必要があります。モジュールはこれを受けてEtherCATの状態をSafeOpに変更し、ALStatusCodeレジスタの問題を示します ([アプリケーションステータスレジスタ, ページ 143](#)参照) 。

下記も参照してください。

- 『Anybus CompactCom 40 Software Design Guide』の「SYNC」
- 『Anybus CompactCom 40 Software Design Guide』の「SYNCオブジェクト」

サポートコマンド

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)

拡張

アトリビュートは EtherCAT においては次のように表されます。

#	名前	アクセス	タイプ	デフォルト値	コメント
1	Cycle time	Get/Set	UINT32		ナノ秒で表したアプリケーションのサイクルタイム。オブジェクトエントリ1C32h、サブインデックス2の設定を置き換えます。(SM Output Parameter、サイクルタイム)
2	Output valid	Get/Set	UINT32	0	SYNCイベント発生から出力が有効になる (Output Valid point) までの相対時間 (単位 : ナノ秒) オブジェクトエントリ1C32h、サブインデックス3の設定を置き換えます。(SM Output Parameter、シフトタイム)
3	Input capture	Get/Set	UINT32	0	SYNCイベント発生から入力を取り込まれる (Input Capture point) までの相対時間 (単位 : ナノ秒) オブジェクトエントリ1C33、サブインデックス3の設定を置き換えます。(SM Input Parameter、シフトタイム)
4	Output processing	Get	UINT32		RDPDI割り込みから有効な出力までに必要な最小時間 (単位 : ナノ秒)。 オブジェクトエントリ1C32h、サブインデックス6の値を指定します。(SM Output Parameter、Output計算とコピー時間) EtherCATで提示されるまでのAnybusの遅延がこの値に加算されます。
5	Input processing	Get	UINT32		Input captureからAnybus CompactComモジュールへのライトプロセスデータの書き込みが完了するまでに必要な最大時間 (単位 : ナノ秒)。 オブジェクトエントリ1C33h、サブインデックス6の値を指定します。(SM Input Parameter、Input計算とコピー時間) EtherCATで提示されるまでのAnybusの遅延がこの値に加算されます。
6	Min cycle time	Get	UINT32		アプリケーションがサポートする最小サイクルタイム オブジェクトエントリ1C32h および1C33h、サブインデックス5の値を指定します。(SM Output、SM Input Parameters、最小サイクルタイム)
7	Sync mode	Get/Set	UINT16		同期モードの選択。このアトリビュートは、アトリビュート8のビットを列挙します。 0 : フリーラン (同期なし) 1: DCによる同期 オブジェクトエントリ1C32h および1C33h、サブインデックス1の値を指定します。(SM Output、SM Input Parameters、同期の種類)
8	Supported sync modes	Get	UINT16		アプリケーションがサポートする同期モードのリストです。各ビットは、アトリビュート7のモードに対応する。 ビット0: 1=フリーランをサポート ビット1: 1=DCをサポート オブジェクトエントリ1C32h および1C33h、サブインデックス4の値を指定します。(SM OutputおよびInput Parameters、同期の種類をサポート)

6.5 EtherCATオブジェクト (F5h)

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と、ホストアプリケーションにEtherCAT固有の設定が実装されます。

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。このモジュールは、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、本モジュールはデフォルト値を使用します。

本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。



このオブジェクトのサポートは任意です。このオブジェクトを実装する場合は、範囲1~6のすべてのアトリビュートをサポートすることを強く推奨します。

最終製品がコンFORMANCEテストに合格するには、その製品を提供するベンダーのベンダーIDが設定されている必要があります。

下記も参照してください。

- 『Anybus CompactCom 40 Software Design Guide』の「エラーコード」

サポートコマンド

オブジェクト :	Get_Attribute
	Get_Object_Description
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	タイプ	値
1	Name	Get	CHAR配列	「EtherCAT」
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)

基本

#	名前	アクセス	タイプ	デフォルト値	コメント
1	Vendor ID	Get	UINT32	E000 0018h	オブジェクトエントリ1018hの設定をこれらの値に置き換えます。 (Identity Object(IDオブジェクト)) 注： デフォルトのVendor IDはHMSのセカンダリVendor IDであるため、コンFORMANCEテストツールを実行するときは使用できません。

拡張

#	名前	アクセス	タイプ	デフォルト値	コメント
2	Product Code	Get	UINT32	0000 0036h	オブジェクトエントリ1018hの設定をこれらの値に置き換えます。 (Identity Object(IDオブジェクト))
3	Major revision	Get	UINT16	Major revision	
4	Minor revision	Get	UINT16	Minor revision	
5	Serial Number	Get	UINT32	製造時に割り当てられた一意の番号	
6	Manufacturer Device Name	Get	CHAR配列 (最大64バイト)	「CompactCom 40 EtherCAT」	オブジェクトエントリ1008hを置き換えます (Manufacturer Device Name (メーカーの機器名))
7	Manufacturer Hardware Version	Get	CHAR配列 (最大64バイト)	X.YY (メジャーバージョン.マイナーバージョン)	オブジェクトエントリ1009hの値を指定します (Manufacturer Hardware Version (メーカーのハードウェアバージョン))
8	Manufacturer Software Version	Get	CHAR配列 (最大64バイト)	X.YY.ZZ (メジャーバージョン.マイナーバージョン.ビルド)	オブジェクトエントリ100Ahの値を指定します (Manufacturer Software Version (メーカーのソフトウェアバージョン))
9	ENUM ADIs	Get	UINT16の配列	-	デフォルト設定。ENUMはEtherCAT上でUNSIGNED8に変換されます。このアトリビュートを実装すると、ENUMはバス上でもENUMに変換されます。このアトリビュートには、ENUMで定義された、ソート済みのADIインスタンス番号のリストが含まれていなければなりません。 このアトリビュートを実装する場合、すべてのENUM ADIについて、アプリケーションデータインスタンスのアトリビュート#6 ("Max. Value") も実装してください。このアトリビュートは必須ではありませんが、これを実装すると、バス上のENUM機能の性能が大幅に向上します。
10	Device Type	Get	UINT32	0000 0000h	実装した場合、この値によってオブジェクトエントリ1000h (Device Type) のデフォルト値が置き換えられます。
11	Write PD assembly instance translation	Get	UINT16の配列	空	このアトリビュートは、アセンブリマッピングオブジェクト内のライトPDインスタンスのデフォルトのTxPDOマッピングオブジェクトを変更するためにアプリケーションで使用できます。このアトリビュートはアセンブリマッピングオブジェクトのアトリビュート11、"Write PD Instance List"に対応します。 配列の各インデックスには、"Write PD Instance List"アトリビュートで同じインデックスにあるインスタンスに対して使用されるTxPDOマッピングオブジェクトの番号が格納されます。 有効値: 1A00h - 1BFFh.
12	Read PD assembly instance translation	Get	UINT16の配列	空	このアトリビュートは、アセンブリマッピングオブジェクト内のリードPDインスタンスのデフォルトのTxPDOマッピングオブジェクトを変更するためにアプリケーションで使用できます。このアトリビュートはアセンブリマッピングオブジェクトのアトリビュート12、"Read PD Instance List"に対応します。 配列の各インデックスには、"Read PD Instance List"アトリビュートで同じインデックスにあるインスタンスに対して使用されるRxPDOマッピングオブジェクトの番号が格納されます。 有効値: 1600h - 17FFh.

#	名前	アクセス	タイプ	デフォルト値	コメント
13	ADI translation	Get	構造体の配列 { UINT16 UINT16 }	空	このアトリビュートをアプリケーションで使用して、通信プロファイル固有のCoEオブジェクト領域 (1000h ~ 1FFFFh) にあるオブジェクトを実装できます。モジュールにすでに実装されているオブジェクトをADIによって置き換えることはできません。このアトリビュートは、2つのUINT16から成るバック構造体の配列として実装されています。最初のUINT16にはADIインスタンス番号を指定し、2番目のUINT16にはそのADIに対応付けるオブジェクトインデックスを指定します。 ADI変換の例、ページ 87 参照。
14	(Reserved)	-	-	-	(将来のために予約)
15	Object subindex translation	Get	構造体の配列 { UINT16 UINT16 UINT8 }	空	このアトリビュートをアプリケーションで使用して、プロファイル固有のCoEオブジェクト領域 (0x1000 ~ 0x1FFF) にあるオブジェクトのサブインデックスを実装できます。モジュールにすでに実装されているサブインデックスをADIによって置き換えることはできません。このアトリビュートは、モジュールで拡張可能と明示的に定義されているオブジェクトにサブインデックスを追加する場合にのみ使用できます。このアトリビュートは、2つのUINT16と1つのUINT8から成るバック構造体の配列として実装されています。最初のUINT16にはADIインスタンス番号を指定し、2番目のUINT16にはそのADIに対応付けるオブジェクトインデックスを指定します。UINT8には、ADIが対応しなければならない後者のオブジェクトのサブインデックスを指定します。オブジェクトディクショナリのインデックス/サブインデックスエントリはVAR型のADIにのみ変換できます。エントリをARRAY型またはRECORD型のADIに変換することはサポートされていません。 オブジェクトサブインデックス変換の例、ページ 87 参照。
16	Enable FoE	Get	BOOL	TRUE (=1)	このアトリビュートは、File access over EtherCATに関連する機能を有効化または無効化します。FoEが無効になっている場合、EtherCATを介してファームウェアをアップグレードしたり、EtherCATを介して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) にアクセスしたりすることはできません。Anybus IPはFoEに対応していません。
17	Enable EoE	Get	BOOL	TRUE (=1)	Ethernet over EtherCATに関連する機能を有効化または無効化します。EoEが無効になっている場合、モジュールはEoE型のメールボックス要求を受け付けず、モジュール内のIT機能は使用不可能となります。
18	Change shift reg switch functionality	Get	BOOL	FALSE (=0)	通常のシフトレジスタ動作モード実行時では、スイッチ1がIPアドレスの最後のオクテットに使用され、スイッチ2がDevice IDに使用されます。このアトリビュートをTRUEに設定するとこの挙動が変更され、スイッチ1がDevice IDに、スイッチ2がIPアドレスの最後のオクテットに使用されるようになります。
19	Set Device ID as Configured station alias	Get	BOOL	FALSE (=0)	通常、設定ステーションエイリアスの値は、EtherCATコンフィグレーションツールからのみ設定できます。これは、このアトリビュートがFalseになっているか実装されていない場合に該当します。このアトリビュートをTrueに設定すると、ネットワークコンフィグレーションオブジェクト (Device ID) のインスタンス1に設定された値が、設定ステーションエイリアス (レジスタ 0x0012とEEPROMの両方) にも設定されます。
20	EtherCAT state	Set	UINT8	1 (=INIT)	CompactComのEtherCATの状態が変わるたびに、モジュールはこのアトリビュートに新しい状態を書き込みます。 1: INIT 2: PRE-OPERATIONAL 3: BOOT 4: SAFE-OPERATIONAL 8: OPERATIONAL 注：このアトリビュートはアサイクリックメッセージチャネルを使用して設定されるため、プロセスデータの妥当性などを点検する場合は信頼できません。このような情報にはCompactComの状態を使用してください。 注：このアトリビュートに対する書き込みは情報提供を行うだけであり、ホストアプリケーションがSet_Attribute要求を受けてエラーを返すことによって状態遷移をNAKにすることはできません。
21	State transition timeouts	Get	UINT32[4] の配列	[1000, 5000, 1000, 200]	このアトリビュートは、EtherCATの状態遷移タイムアウトを変更するために実装できます。アトリビュートは4つの要素を持つUINT32の配列であり、各要素は以下を意味します。 0：ミリ秒で表したPreopTimeout。INIT->PREOPおよびINIT->BOOT遷移のタイムアウト。ESI要素のPreopTimeoutに相当します。 1：ミリ秒で表したSafeopOpTimeout。SAFEOP->OPおよびPREOP->SAFEOP遷移のタイムアウト。ESI要素のSafeopOpTimeoutに相当します。 2：ミリ秒で表したBackToInitTimeout。すべての状態遷移がINIT状態に戻るときに適用されるタイムアウト。ESI要素のBackToInitTimeoutに相当します。

#	名前	アクセス	タイプ	デフォルト値	コメント
					3: ミリ秒で表したBackToSafeopTimeout. OP->SAFEOP遷移のタイムアウト。ESI要素のBackToSafeopTimeoutに相当します。 状態遷移タイムアウトの詳細については、以下のセクションを参照してください
22	Compare identity lists	Get	BOOL	TRUE (=1)	このアトリビュートは、モジュラーデバイスプロファイルを使用する場合にのみ関係があります。 Falseに設定されている場合、モジュールはPREOPからSAFEOPへの遷移時に、検出された設定済みモジュールのIDリストを比較しません。
23	FSoE status indicator	Set	ENUM	255 (= 状態非表示)	このアトリビュートは、Safety over EtherCATが有効な場合にのみ関係があります。 このアトリビュートは、FSoE状態LEDを更新するためにホストによって使用されます。CompactComモジュールが安全モジュールからLEDの状態変化を取得すると、このアトリビュートが更新されます。 0 = オフ 1 = 点滅 2 = オン 3 = 1回点滅 4 = 早い点滅 5 = 1回点滅、早い点滅 6 = 2回点滅、早い点滅 7 = 3回点滅、早い点滅 8 = 4回点滅、早い点滅 9 = 5回点滅、早い点滅 10 = 6回点滅、早い点滅 254 = フェイルセーフ状態 255 = 状態非表示。
24	Clear IdentALSts	Get	BOOL	FALSE (=0)	このアトリビュートを実装して1 (= True) に設定すると、CompactComは常にEEPROM一般カテゴリのIdentALStsビットをクリアします。CompactComはこのように、Explicit Device IDのサポートを示しません。 アトリビュートが実装されていない場合、または0 (= False) に設定されている場合は、ネットワークコンフィグレーションオブジェクトのインスタンス1 (Device ID) のValueアトリビュートが書き込まれると、すぐにEEPROM一般カテゴリのIdentALStsフラグが1になります。 通常、このアトリビュートのサポートが必要なのは、シフトレジスタ動作モードを使用していて、Device IDスイッチを持たないホストアプリケーションのみです。
25	SII Order Number	Get	CHAR配列 (最大64バイト)	オブジェクト1008hの値、メーカーのデバイス名 (上記のアトリビュート#6 を参照) 。	このアトリビュートを実装する必要があるのは、SII Order Number文字列で報告するために、ホストアプリケーションがCoEオブジェクト0x1008h以外の値を必要とする場合のみです。
26	SII Device Name	Get	CHAR配列 (最大64バイト)	オブジェクト1008hの値、メーカーのデバイス名 (上記のアトリビュート#6 を参照) 。	このアトリビュートを実装する必要があるのは、SII Device Name Information文字列で報告するために、ホストアプリケーションがCoEオブジェクト0x1008h以外の値を必要とする場合のみです。
27	Last FoE Data ACK delay	Get	UINT16	500	FoEファイル書き込み転送時の最後のFoE Data要求に対して、Anybus CompactCom 40 EtherCATがACKを遅延させる時間をミリ秒で表したもの。 注： この遅延時間が経過する前に、ABCCファームウェア候補領域に書き込めるようファイルが開いた場合、Anybus CompactComは直ちにACKを送信します。 この遅延はSEMIデバイスプロファイルのファームウェアアップグレード仕様に対応するアプリケーション用ですが、FoE転送のタイムアウトが非常に短く設定されているEtherCATマスターのコンフィグレーションで使用するアプリケーションにも使用できます。

アトリビュート#21： State transition timeouts

通常は、デフォルトの状態遷移タイムアウトを使用できます。しかし、ホストアプリケーションが状態遷移のタイムアウトを変更する必要がある状況がいくつかあります。

- アプリケーションがプロセスデータの再マッピングコマンドを実行するのに長い時間を要する場合は、SafeOpTimeoutの値を大きくする必要があります。EtherCAT側からの再マッピングは、PREOP->SAFEOP遷移時に行われます。デフォルトのタイムアウト（5秒）を超過するリスクがある場合は、このタイムアウトの値を大きくする必要があります。
- ホストアプリケーションが同期的に動作し、SYNC信号などにロックするのに長時間を要する場合、SafeOp timeoutの値を大きくする必要があります。
- ホストアプリケーションが、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を使用したEtherCATでのファームウェアアップグレードに対応している場合、BackToInitTimeoutの値を大きくする必要があります。これは、ホストアプリケーションがHIFFファイルの検証を行う必要があり、BOOT->INIT遷移で同ファイルをAnybusファイルシステムインターフェースオブジェクトにアップロードした場合などに必要です。EtherCATを介したファームウェアのアップグレードについては、EtherCAT Technology Groupの「SEMI Device FW Upgrade」（SEMIデバイスFWアップグレード）を参照してください。

これらのタイムアウトの値を変更した場合は、新しいタイムアウトを使用してESIファイルを更新する必要があります。EtherCAT Technology Groupの「EtherCAT Slave Information Specification」（EtherCATスレーブ情報仕様、表71）を参照してください。

コマンドの詳細： Get_Object_Description

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

ホストアプリケーションによるこのコマンドのサポートは任意で、"Get Object Description" SDO情報応答において、CompactComによって追加されたADI構造体に関する情報を変更するために使用されます。

通常、ADI構造体は、EtherCATデータ型0x2A (事前定義のRECORDなし) を持つRECORDオブジェクトに変換されます。

このコマンドのサポートを実装することで、ホストアプリケーションは、ADI構造体に対応するオブジェクトに対して返されるオブジェクトコードとオブジェクトデータ型の両方を変更できます。

これは、一部のEtherCATプロファイル (一部のSEMIデバイスプロファイルなど) を実装する場合に必要となります。

ホストアプリケーションがこのコマンドでエラーコード0x03 (Unsupported object)、0x04 (Unsupported instance)、または0x05 (Unsupported command) のいずれかで応答した場合、CompactComはその後のGet Object Description要求にこのコマンドを使用しません。

• コマンド詳細

フィールド	内容	コメント
CmdExt[0]	オブジェクトインデックス、下位バイト	これは、Get Object Description SDO情報要求を受け取ったオブジェクトインデックスです。
CmdExt[1]	オブジェクトインデックス、上位バイト	

• 応答詳細

フィールド	内容	コメント
CmdExt[0]	オブジェクトインデックス、下位バイト (コマンドのミラー)	これは、Get Object Description SDO情報要求を受け取ったオブジェクトインデックスです。
CmdExt[1]	オブジェクトインデックス、上位バイト (コマンドのミラー)	
Data[0-1]	EtherCATデータ型 (UINT16)	提供されたオブジェクトインデックスに対するGet Object Description応答で報告されるEtherCATデータタイプ。
Data[2]	EtherCATオブジェクトコード (UINT8)	提供されたオブジェクトインデックスに対するGet Object Description応答で報告されるEtherCATオブジェクトコード。

ADI変換の例

診断オブジェクト (10F3h) とタイムスタンプオブジェクト (10F8h) をホストアプリケーションに実装するとします。そのためには、これらのオブジェクトのCoE実装に対応する2つのADIを作成する必要があります。例えば、診断オブジェクトに対してADI F0F3hを作成し、タイムスタンプオブジェクトに対してF0F8を作成します。さらに、ADI translationアトリビュートに対して次のデータを実装する必要があります。

例 2:

```
[
  {
    F0F3h
    10F3h
  }
  {
    F0F8h
    10F8h
  }
]
```

これで、これらのCoEオブジェクトに対するSDO要求は、対応するADIに転送されます。このアトリビュートで設定されたCoEオブジェクトがモジュールによって実装されている場合、そのオブジェクトに対するすべての要求はモジュール自体によって処理され、ホストアプリケーションには何も転送されません。

オブジェクトサブインデックス変換の例

0x1C32および0x1C33オブジェクトの同期エラーサブインデックス (サブインデックス32) をホストアプリケーションに実装するとします。そのためには、これらのエントリのCoE実装に対応する2つのADIを作成する必要があります。エントリ0x1C32:32に対してADI 0xF0FDが、エントリ0x1C33:32に対してADI 0xF0FEが作成されるとします。この場合、「オブジェクトサブインデックス変換」アトリビュートに次のデータを実装する必要があります。

例 3:

```
[
  {
    0xF0FD
    0x1C32
    32
  }
  {
    0xF0FE
    0x1C33
    32
  }
]
```

これで、これらのCoEオブジェクト/サブインデックスエントリに対するSDO要求は、対応するADIに転送されます。

このアトリビュートで設定されたCoEエントリがモジュールによって実装されている場合、そのエントリに対するすべての要求は (オブジェクトが追加のサブインデックスによる拡張をサポートしていない場合と同じように) モジュール自体によって処理され、ホストアプリケーションには何も転送されません。

6.6 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配列	「Ethernet」
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 (トランスベ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 (トランスベ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/無効 (トランスベ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。)
7	Enable admin mode	Get	BOOL	False (無効)	FTP adminモードの有効/無効 (トランスベ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。 ブレプログラム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。 ブレプログラム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。 このアトリビュートはPROFINETおよびイーサネットPOWERLINKでは使用されません 00h: 有効 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）。
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。 このアトリビュートはPROFINETでは使用されません 00h: 有効 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）。 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EtherNet/IPおよびModbus-TCPデバイスで利用可能です。
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。コンフィグレーションスイッチの値が1～245に設定されている場合、スタンダロンのシフトレジスタモードで使用されます。この場合、IPアドレスは次のように設定されます。 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はコンフィグ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コンフィグ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)のコンフィグ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デバイスについて有効です。 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

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 ⁸ 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には不使用
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にはTypeフィールドとClient Identifierが含まれており、RFC 2132の定義に準拠します。アトリビュート#23を介してモジュールへ送ることのできる最大長は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。

7 Webサーバ

7.1 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サーバはイーサネットホストオブジェクト (F9h) で完全に無効にできます。

Webサーバは最大20の同時接続をサポートでき、ポート80を介して通信を行います。

下記も参照してください。

- [FTPサーバ, ページ 98](#)
- [Server Side Include \(SSI\), ページ 101](#)
- [JSON, ページ 120](#)
- [Ethernetホストオブジェクト \(F9h \), ページ 88](#)

7.2 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 フォルダー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み取り専用であり、削除も上書もできません。Webサーバはまず Webのrootフォルダー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 フォルダーにあるファイルを検索し、ファイルがあたかもWebの root フォルダーにあるように表示します。vfs フォルダーのデフォルト ファイルと全く同じ名前でWebのroot フォルダー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フォルダー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。

7.2.1 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設定を変更するインターフェースを提供します。

The figure shows two screenshots of a web-based configuration interface. The top screenshot is titled 'IP Configuration' and features a sidebar with a 'MODULE' menu containing 'Overview', 'Parameters', 'NETWORK', 'Status', 'Configuration', 'SERVICES', and 'SMTP'. The main content area has a 'DHCP' section with an 'Enabled' dropdown menu. Below this are input fields for 'IP Address', 'Subnet Mask', and 'Gateway Address', all with '0.0.0.0' as the default value. Further down are fields for 'Host Name', 'Domain name', 'DNS Server #1', and 'DNS Server #2', each with '0.0.0.0' as the default. A 'Save settings' button is at the bottom. The bottom screenshot is titled 'SMTP configuration' and has a similar sidebar. The main content area has input fields for 'Server:', 'User:', 'Password:', and 'Confirm password:'. A 'Save settings' button is also present.

図 3

変更を反映させるにはモジュールをリセットする必要があります。

使用可能なIPコンフィグレーションの設定

Name	説明
DHCP	DHCPを有効化または無効化するためのチェックボックス デフォルト値： 無効
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 Address	
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

使用可能なSMTP設定

Name	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

7.2.2 ネットワーク状態ページ

ネットワーク状態Webページには以下の情報が含まれます。

Current IP Configuration	説明
DHCP:	-
Host Name:	-
IP Address:	-
Subnet Mask:	-
Gateway Address:	-
DNS Server #1:	-
DNS Server #2:	-
Domain Name:	-
Current Ethernet Status	説明
MAC Address	-
Port 1	現在のリンク速度とduplex設定。
Port 2	現在のリンク速度とduplex設定。
EtherCAT Statistics	説明
Logical EoE port link:	EoEを使用した通信を可能にするには、物理的なイーサネットポートの1つにリンクがあるだけでは十分ではありません。また、ネットワーク状態が少なくともPRE-OPERATIONALの状態にあり、メールボックス通信がアクティブになっている必要があります。「Logical EoE port link」は、デバイスがEoEフレームを送受信できるようにするためにこれらの要件が満たされているかどうかを示します。
Invalid frame counter IN port:	ESCレジスタ0x300の値。受信した無効なフレームの数。
Rx error counter IN port:	ESCレジスタ0x301の値。PHYによって報告されたRxエラーの数。
Forwarded error counter IN port:	ESCレジスタ0x308の値。転送されたRxエラーの数。
Lost link counter IN port:	ESCレジスタ0x310の値。ポートでリンクが失われた回数。
Invalid frame counter OUT port:	ESCレジスタ0x302の値。受信した無効なフレームの数。
Rx error counter OUT port:	ESCレジスタ0x303の値。PHYによって報告されたRxエラーの数。
Forwarded error counter OUT port:	ESCレジスタ0x309の値。転送されたRxエラーの数。
Lost link counter OUT port:	ESCレジスタ0x311の値。ポートでリンクが失われた回数。
EoE 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:	受信した非ユニキャストパケット数 (ブロードキャストおよびマルチキャスト)。
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 (ブロードキャストおよびマルチキャスト)。
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。

7.3 サーバコンフィグレーション

7.3.1 基本情報

基本的なWebサーバのコンフィグレーション設定は、システムファイル\http.cfgに格納されます。このファイルには、Webインターフェース用root 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:
[WebRoot]
\web

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

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

Web Root Directory [WebRoot]	Webサーバはこのディレクトリの外にあるファイルにアクセスすることはできません。
Content Types [FileTypes]	ファイル拡張子とその報告されたコンテンツタイプのリスト。 下記も参照してください。 デフォルトのコンテンツタイプ, ページ 96
SSI File Types [SSIFileTypes]	デフォルトでは拡張子 'shtml' のファイルだけがSSI用にスキャンされます。追加のSSI ファイルタイプを必要に応じてここに追加できます。

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。

7.3.2 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, ページ 92](#)

7.3.3 デフォルトのコンテンツタイプ

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。

7.3.4 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から読込まれます。

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

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

8 FTPサーバ

8.1 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。

8.2 ユーザーアカウント

ユーザーアカウントは、コンフィグレーションファイル\ftp.cfgに格納されます。このファイルには、すべての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含まれています。ユーザーは自分のホームディレクトリの外にあるファイルにアクセスすることはできません。

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、どのユーザー名/パスワードの組合せでも承認され、homeディレクトリはFTP root (\ftp\) になります。
- ユーザーがログインできるようにするには、ユーザーのホームディレクトリもファイルシステムに存在する必要があります。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モードがイーサネット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 フォルダー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）。

8.3 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記入します
3. **Enter**を押します。 エクスプローラ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。接続に成功するとファイルシステムはエクスプローラウィンドウに表示されます。

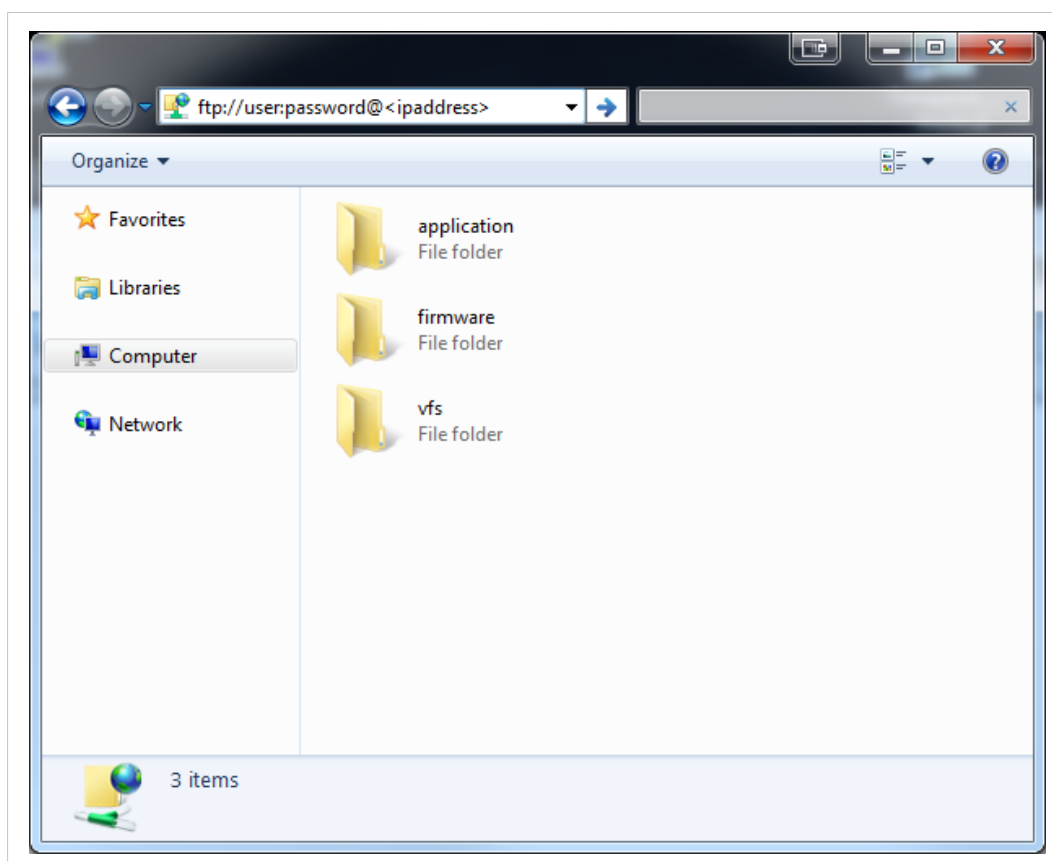


図 4

9 電子メールクライアント

9.1 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。

9.2 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。

10 Server Side Include (SSI)

10.1 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 'shtml' のファイルだけがSSI用にスキャンされます。

10.2 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)

10.3 コマンドの機能

10.3.1 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



「*command*」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

10.3.2 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 『filename』」
キーが見つからない	「Tag (key) not found」

例

次のSSI...

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

... 次のファイル ('\example.cnf') と併用...

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

... 文字列 'Third' が返されます。

10.3.3 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、その後に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 『filename』」
認証エラー	「Authentication error」
ファイルオープンエラー	「Failed to open file 『filename』」
ファイル書き込みエラー	「Could not store configuration to 『filename』」

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。

10.3.4 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 119](#)

10.3.5 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)

10.3.6 ChangeLanguage()

このコマンドはHTMLフォームオブジェクトに基づく言語設定を変更します。



この機能は電子メールメッセージで使用できません。

構文:

```
<!--#exec cmd_argument='ChangeLanguage ( "source" )'-->
```

source:

新たな言語設定を含むフォームオブジェクト名。

渡される値は次のように一桁でなければなりません:

フォーム値	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

デフォルト出力

シナリオ	デフォルト出力
成功	「Language changed」
エラー	「Failed to change language」

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、ChangeLanguage()コマンドで評価されます。

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

<!--#exec cmd_argument='ChangeLanguage ("lang")'-->

<FORM action="test.shtml">
  <P>
    <LABEL for="lang">Language (0-4) : </LABEL><BR>
    <INPUT type="text" name="lang"><BR><BR>

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

</BODY>
</HTML>
```



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

10.3.7 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。改行文字 (「
」など) を指定してください。

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 『filename』」

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：

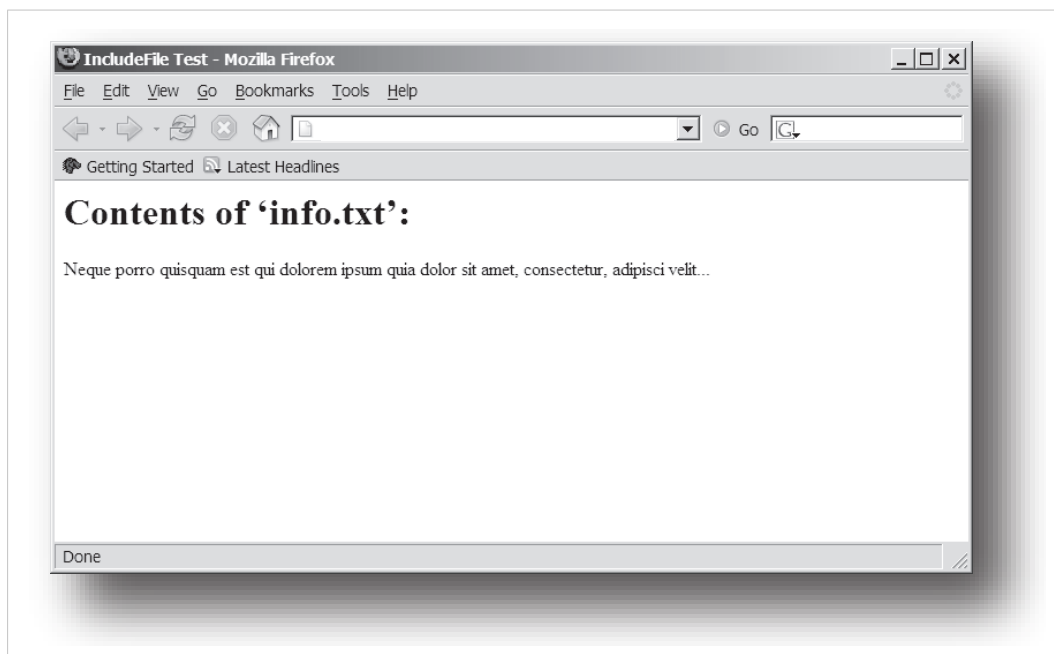


図 5

下記も参照してください。

- [Include File, ページ 101](#)

10.3.8 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 『filename』」
認証エラー	「Authentication error」
ファイル書き込みエラー	「Could not store configuration to 『filename』」

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。

10.3.9 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、下記の「Formatting Tags」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。
- [ABCCMessage\(\)](#), ページ 115

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 118)

例

以下を参照してください。

- [ABCCMessage\(\)](#), ページ 115
- [例 \(Get_Attribute \) ;](#), ページ 117

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の接頭文字が付けます。

- 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。 結果が大きいても、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

10.3.10 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、これが違うと結果は未定義です。 詳細については、下記の「Formatting Tags」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 115

デフォルト出力

シナリオ	デフォルト出力
成功	「Success」
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 118)

例

下記も参照してください。

[ABCCMessage\(\)](#), ページ 115

例 ([Set_Attribute\(\)](#)), ページ 117

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	SINT8/UINT8 SINT16/UINT16 SINT32/UINT32

タイプ	入力	引数データ型
	1～9: 10進数	
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: <ul style="list-style-type: none"> 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

10.4 引数関数

10.4.1 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()	-

10.4.2 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。

数値入力に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	不可	「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：「We usually refer to it as \『the Egg\』」
FLOAT	はい	5.6538e2
NONE	No	コマンドはデータを保持できないので、データ型也没有せん

- 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値を読み取ります。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せず、データ型がありません



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（不正なGet_Attribute要求など。この場合メッセージのデータは直前に不正であったときでもオブジェクトへ無修正のまま渡される）を確認しません。このことを守らないとデータやその他の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IPアドレスをscanf()とABCCMessage()で設定する方法を示します。(scanf () によって解析された) 渡された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値「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 119](#)

10.5 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:

<pre>[ABCCMessage_X] 0: 「Success string」 1: 「Error string 1」 2: 「Error string 2」 ... 16 「Error string 16」</pre>	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 118 参照。
<pre>[GetConfigItem_X] 0: 「Success string」 1: 「Authentication error string」 2: 「File open error string」 3: 「Tag not found string」</pre>	「%s」を使用して、ファイル名を含めます。
<pre>[SetConfigItem_X] 0: 「Success string」 1: 「Authentication error string」 2: 「File open error string」 3: 「File write error string」</pre>	「%s」を使用して、ファイル名を含めます。
<pre>[IncludeFile_X] 0: 「Success string」 1: 「Authentication error string」 2: 「File read error string」</pre>	「%s」を使用して、ファイル名を含めます。
<pre>[scanf_X] 0: 「Success string」 1: 「Parsing error string」</pre>	-
<pre>[ChangeLanguage_X] 0: 「Success string」 1: 「Change error string」</pre>	-

上記のすべてのコンテンツは、異なる言語の各タグの値「X」を変更して、ファイルに複数回含めることができます。モジュールは次に正しい出力文字列を言語設定に基づいて選択します。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 105

11 JSON

11.1 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。

11.1.1 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。

11.1.2 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 Server」のセクションで説明しています。

11.1.3 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。

11.2 JSONオブジェクト

11.2.1 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「order number」です。 最初に実装されたADIは常にオーダー番号0を取得します。<count>個の既存のADI値が返されます。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「order number」です。 最初に実装されたADIは常にオーダー番号0を取得します。<count>個の既存のADIの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	-
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 139 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 139 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「order number」です。 最初に実装されたADIは常にオーダー番号0を取得します。<count>個の既存のADIの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	-
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 139 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 139 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 139 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 139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
}
```

11.2.2 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	[high, low] ミリ秒 (ms)
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": "0085",
}
```

11.2.3 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: Link
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構成を変更するには、network/ipconf.jsonを使用してください。上記の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アクティブ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、GET要求に引数として追加することで構成可能です。

```
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、Anybusエラーコードを用いた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
innucast	番号	IN: ブロードキャストおよびマルチキャストパケット
indiscards	番号	IN: 破棄パケット
inerrors	番号	IN: エラー
inunknown	番号	IN: サポートされていないプロトコルタイプ
outoctets	番号	OUT: バイト
outucast	番号	OUT: ユニキャストパケット
outnucast	番号	OUT: ブロードキャストおよびマルチキャストパケット
outdiscards	番号	OUT: 破棄パケット
outerrors	番号	OUT: エラー

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": "bacnetaplserversstats",
      "title": "BACnet Application Layer Server Statistics" },
    { "identifier": "bacnetaplcclientstats",
      "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”値です) :

```
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 },
]
```

「bacnetaplserverstats」

```
[
  { "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 },
]
```

「eplifcounters」

```
[
  { "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 },
]
```

「ectstats」

```
[
  { "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 },
]
```

「pnprof」

```
[
  { "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" }
]
```

11.2.4 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
サーバ	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
ユーザー	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],
]
```

11.2.5 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
```

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

A 機能の分類

Anybus CompactComとアプリケーションのアトリビュートやサービスを含むオブジェクトは、次の二つのカテゴリに分けられます。基本および拡張。

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、アトリビュート、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。EtherCATでは、この機能はシンクマネージャウォッチドッグに割り当てられています。このウォッチドッグを使用すると、マスターとの通信の喪失を検出できます。シンクマネージャウォッチドッグはマスターが有効にします。

EtherCAT固有の解釈：

SUPビット	解釈
0	シンクマネージャウォッチドッグが無効になっているか、または動作していません。
1	シンクマネージャウォッチドッグが有効になっており、ウォッチドッグが動作しています。



リードプロセスデータのサイズがゼロの場合、ウォッチドッグと監視ビット (SUP) は利用できません。

B.2 Anybusステートマシン

AnybusのステートマシンとEtherCATネットワークの状態との対応を下表に示します。

Anybusの状態	対応するEtherCATの状態
WAIT_PROCESS	INIT, BOOTSTRAP or PRE-OPERATIONAL
ERROR	("AL-Status"の"Error Ind"ビットがセットされている)
PROCESS_ACTIVE	OPERATIONAL
IDLE	SAFE-OPERATIONAL
EXCEPTION	(EtherCATインターフェースがINIT状態に移行し、通信を再開するには電源のオフ/オンが必要なことがマスターに通知される)

B.3 アプリケーションステータスレジスタ

アプリケーションステータスレジスタは、主に、SYNCアプリケーションで使用されます。これは、対象のネットワークがプロセスデータの致命的なエラーをマスターに通知する機能をサポートするアプリケーションで使用されます。この機能がサポートされている場合、Anybus CompactComモジュールは、アプリケーションにより書き込まれた以下のステータスコードを受け付けて処理します。

アプリケーションがアプリケーションステータスレジスタにエラーステータスを設定した場合、モジュールはEtherCATの状態をSafeOpに設定します。値は次表のように変換されて、ALStatusCodeレジスタに格納されます。

値	エラー	ALStatusCode : ALSTATUSCODE_XXX (#)	コメント
0000h	エラーなし	-	アプリケーションはPROCESS_ACTIVE状態で動作できます
0001h	同期していない	NOSYNCEERROR (002Dh)	アプリケーションはSYNC信号に同期しておらず、PROCESS_ACTIVEに移行する準備ができていません。
0002h	シンクコンフィグレーションエラー	INVALIDSYNCCFG (0030h)	SYNCホストオブジェクトのコンフィグレーションに問題があるため、アプリケーションがPROCESS_ACTIVEに移行できません。
0003h	リードプロセスデータコンフィグレーションエラー	INVALIDOUTPUTMAPPING (0025h)	現在のリードプロセスデータのマッピングに問題があるため、アプリケーションがPROCESS_ACTIVEに移行できません。
0004h	ライトプロセスデータコンフィグレーションエラー	INVALIDINPUTMAPPING (0024h)	現在のライトプロセスデータのマッピングに問題があるため、アプリケーションがPROCESS_ACTIVEに移行できません。
0005h	同期喪失	FATALSYNCEERROR (002Ch)	アプリケーションが同期へのロックを喪失しました。モジュールがPROCESS_ACTIVE状態にある場合、ERRORに移行します。
0006h	過度のデータ消失	NOVALIDINPUTSANDOUTPUTS (002Bh)	ネットワークからのプロセスデータフレームが大量に失われていることをアプリケーションが検出しました。モジュールがPROCESS_ACTIVE状態にある場合、ERRORに移行します。
0007h	出力エラー	DCSYNCEIOERROR (0033h)	アプリケーションに問題があるため、出力を実行できません。モジュールがPROCESS_ACTIVE状態にある場合、ERRORに移行します。

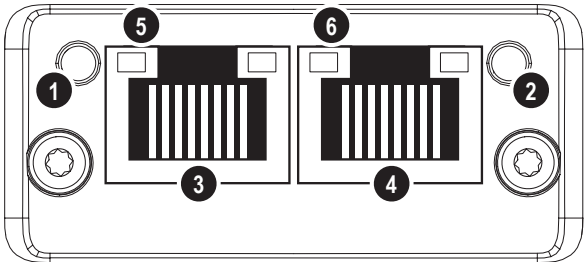
B.4 Application Watchdog Timeout Handling

アプリケーションウォッチドッグがタイムアウトした場合、Anybus CompactCom 40 EtherCATモジュールはEXCEPTION状態になります。

C 技術仕様

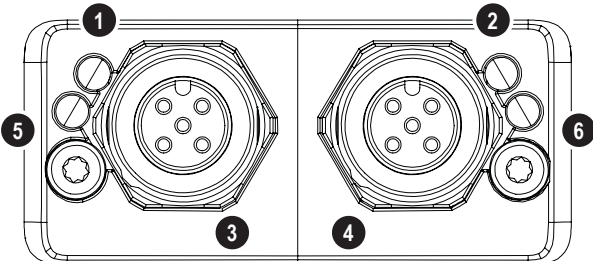
C.1 前面図

C.1.1 前面図 (RJ45コネクタ)

#	項目	
1	RUN LED	
2	ERROR LED	
3	EtherCAT (INポート)	
4	EtherCAT (OUTポート)	
5	リンク/アクティビティ (INポート)	
6	リンク/アクティビティ (OUTポート)	

RUN LEDおよびERROR LEDの点滅シーケンスは、ETG1300_S_R_V1i1i0_IndicatorLabelingSpecification.pdf (ETG) で定義されています。

C.1.2 前面図 (M12コネクタ)

#	項目	
1	RUN LED	
2	ERROR LED	
3	EtherCAT (INポート)	
4	EtherCAT (OUTポート)	
5	リンク/アクティビティ (INポート)	
6	リンク/アクティビティ (OUTポート)	

RUN LEDおよびERROR LEDの点滅シーケンスは、ETG1300_S_R_V1i1i0_IndicatorLabelingSpecification.pdf (ETG) で定義されています。

C.1.3 RUN LED

このLEDはEtherCATデバイスの状態を表します。

LEDの状態	意味	説明
オフ	INIT	EtherCAT機器は"INIT"状態にあります (または電源オフ)
緑	OPERATIONAL	EtherCAT機器は"OPERATIONAL"状態にあります
緑点滅	PRE-OPERATIONAL	EtherCAT機器は"PRE-OPERATIONAL"状態にあります
緑1回点滅	SAFE-OPERATIONAL	EtherCAT機器は"SAFE-OPERATIONAL"状態にあります
早い点滅	BOOT	EtherCAT機器は"BOOT"状態にあります
赤	(致命的な事象)	RUN LEDとERR LEDが赤点灯した場合、致命的な事象が発生していることを示します。このとき、バスインターフェースは物理的にパッシブな状態になります。HMSの技術サポートにお問い合わせください。

C.1.4 ERR LED

このLEDはEtherCATの通信エラーなどを表します。

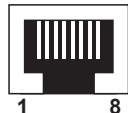
LEDの状態	意味	説明
オフ	エラーなし	エラーは発生していません (または電源オフ)
赤点滅	無効なコンフィグレーション	レジスタまたはオブジェクトの設定が無効なため、マスターから送信された状態に変更できません。
赤1回点滅	自律的な状態変化	スレーブ機器のアプリケーションがEtherCATの状態を自律的に変更しました。
赤2回点滅	シンクマネージャウォッチドッグがタイムアウトしました	詳細は ウォッチドッグ機能 , ページ 15を参照してください。
赤	アプリケーションコントローラーの故障	AnybusモジュールがEXCEPTION状態にあります。 RUN LEDとERR LEDが赤点灯した場合、致命的な事象が発生していることを示します。このとき、バスインターフェースは物理的にバッドな状態になります。HMSの技術サポートにお問い合わせください。
早い点滅	ブート中にエラーが発生	例えば、ファームウェアのダウンロードが失敗した場合など。

C.1.5 リンク/アクティビティ

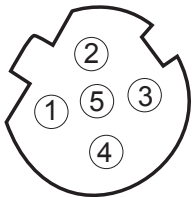
このLEDは、EtherCATのリンク状態とアクティビティを表します。

LEDの状態	意味	説明
オフ	リンクなし	リンクが検出されていません (または電源オフ)
緑	リンク検出。アクティビティなし	リンクを検出しましたが、トラフィックが検出されていません
緑、はやい点滅	アクティビティあり	リンクを検出し、トラフィックを検出しました

C.1.6 イーサネットコネクタ (RJ45)

ピン	信号	注意事項	
1	Tx+	-	
2	Tx-	-	
3	Rx+	-	
4	-	通常は使用しません。信号の整合性を保つため、これらのピンを相互に接続し、モジュールのフィルター回路を介してPEに終端します。	
5	-		
6	Rx-	-	
7	-	通常は使用しません。信号の整合性を保つため、これらのピンを相互に接続し、モジュールのフィルター回路を介してPEに終端します。	
8	-		

C.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	

C.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、モジュールは全般的な『Anybus CompactCom M40 Hardware Design Guide』に記載されているとおり、FE pad / FE機構を使用してファンクショナルアースに正しく接続してください。

これらのFE要件が満たされていない場合、HMS Industrial Networks ABは適正なEMC動作が保証されません。

C.3 電源

C.3.1 電源電圧

本モジュールを使用するには、全般的な『Anybus CompactCom M40 Hardware Design Guide』で指定された3.3 Vの安定化電源が必要です。

C.3.2 消費電力

Anybus CompactCom 40 EtherCAT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。Anybus CompactComプラットフォームにおける消費電力の分類についての詳細は、『Anybus CompactCom Hardware Design Guide』を参照してください。

現在のハードウェア設計では、最大430 mAを消費します。



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。

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、いかなる場合においてもAnybus CompactCom 40 EtherCATはクラスBモジュールに留まります。

C.4 環境仕様

詳細については、『Anybus CompactCom Hardware M40 Design Guide』を参照してください。

C.5 EMC準拠

詳細については、『Anybus CompactCom Hardware M40 Design Guide』を参照してください。

D タイミングと性能

D.1 基本情報

この章では、Anybus CompactCom 40 EtherCATについて検証/文書化された、タイミングと性能に関するパラメータについて説明します。

以下のタイミングが測定されています。

カテゴリ	パラメータ	ページ
起動時の遅延	T1, T2	147
NW_INITの処理	T100	147
イベントベースのWrMsgビジー時間	T103	147
イベントベースのプロセスデータ遅延時間	T101, T102	148

詳細については、『Anybus CompactCom 40 Software Design Guide』を参照してください。

D.2 内部タイミング

D.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 EtherCAT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）	11	ms
T2	Anybus CompactCom 40 EtherCAT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して処理できる（シリアルモード）	11	ms

D.2.2 NW_INITの処理

このテストは、Anybus CompactCom 40 EtherCAT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	3.6	ms

D.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	2.8	7.2	μs

D.2.4 イベントベースのプロセスデータ遅延時間

「Read process data delay」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割り込みをアサートするまでの時間として定義されます。

「Write process data delay」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、新しい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	-	-	228	ns
T102	ライトプロセスデータ遅延時間	-	-	170	ns

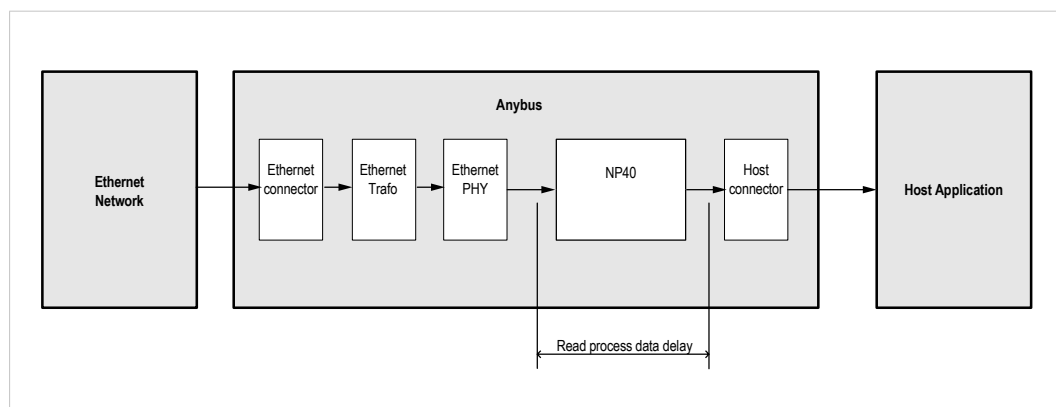


図 6

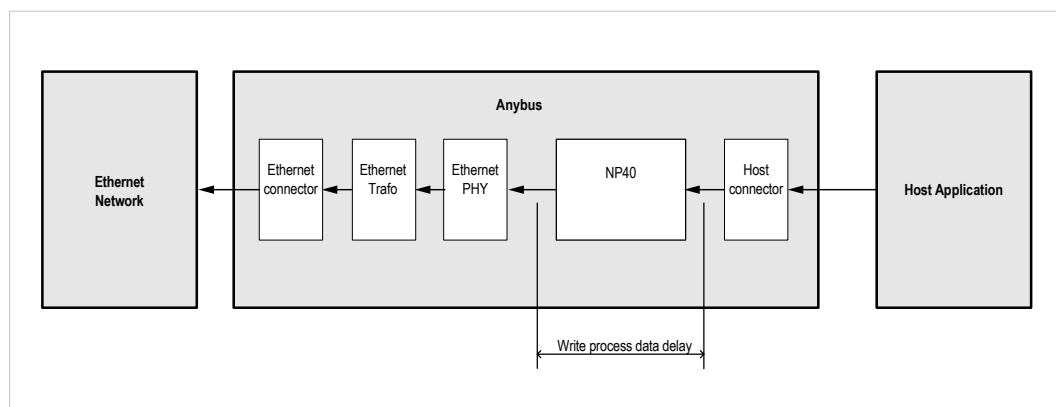


図 7

E 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)

E.1 概要

Anybus CompactCom 40 EtherCAT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) に対応します。Anybus IPconfigはHMSのWebサイトwww.anybus.comから無料でダウンロードできます。このユーティリティは、UDPポート3250を介してネットワークに接続されているあらゆるAnybus製品のネットワーク設定にアクセスするために使用できます。

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。

E.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製品のために自動スキャンされます。**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、リストのエントリー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現れます。

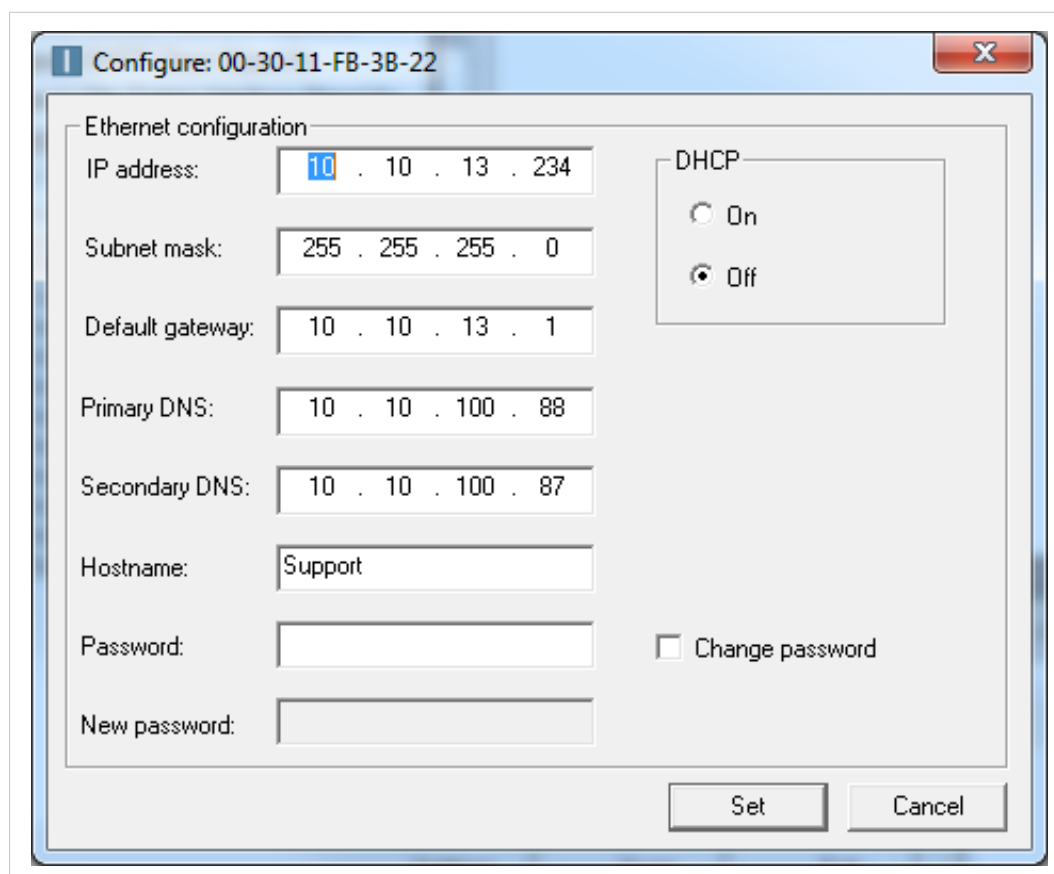


図 8

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、パスワードを使用してコンフィグレーションを許可されていないアクセスから保護することができます。パスワードを入力するには、**Change password**チェックボックスにチェックマークを付けてから、**New password**テキストフィールドにパスワードを入力します。

F 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 EtherCAT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。

F.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。
 - ホストアプリケーションと40シリーズ間の新しい操作手順をサポートするために、新しいドライバーとホストアプリケーションのサンプルコードがwww.anybus.com/starterkit40で入手できます。このドライバーは、30シリーズと40シリーズの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。



本章では、30シリーズと40シリーズとの相違についてのみ扱います。Anybus CompactCom 40シリーズの新規および拡張された機能の説明については、すべてのドキュメントを確認できるサポートページを参照してください。

サポートページへのリンク：www.anybus.com/support

F.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。

F.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 9 Anybus CompactCom M30/M40

F.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。

F.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 AB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 ABまでお問い合わせください。

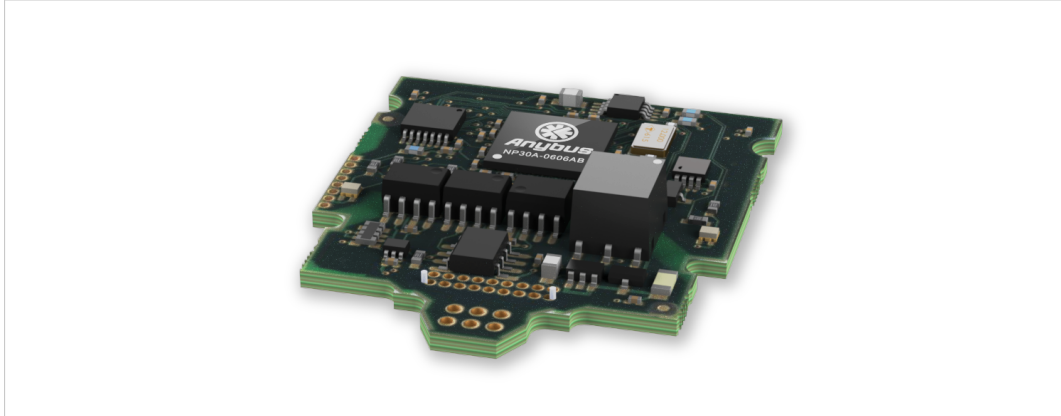


図 10 Anybus CompactCom B30

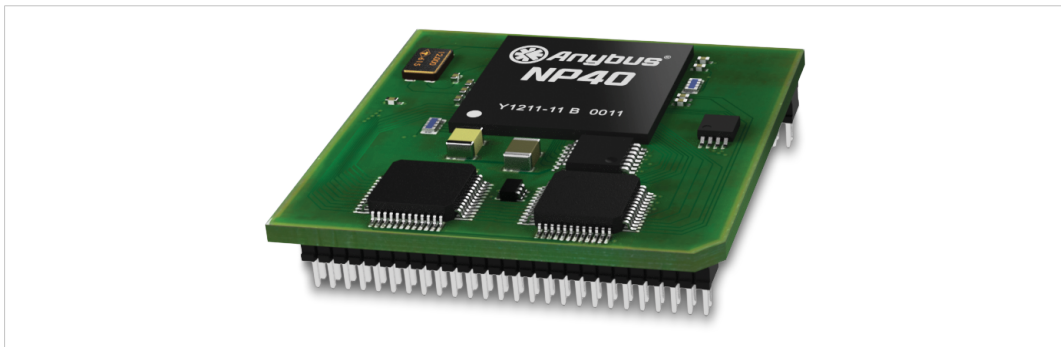


図 11 Anybus CompactCom B40 – 1 (移行用ではありません)

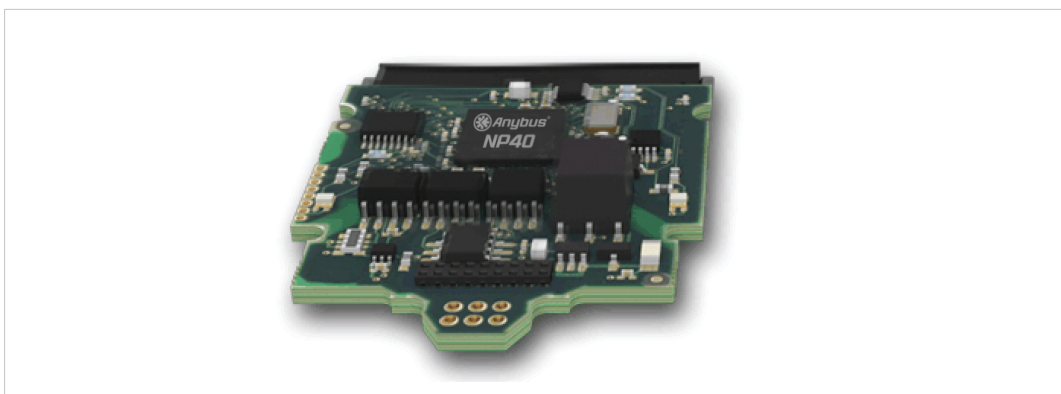


図 12 Anybus CompactCom B40-2

F.2.4 ホストアプリケーションインターフェース

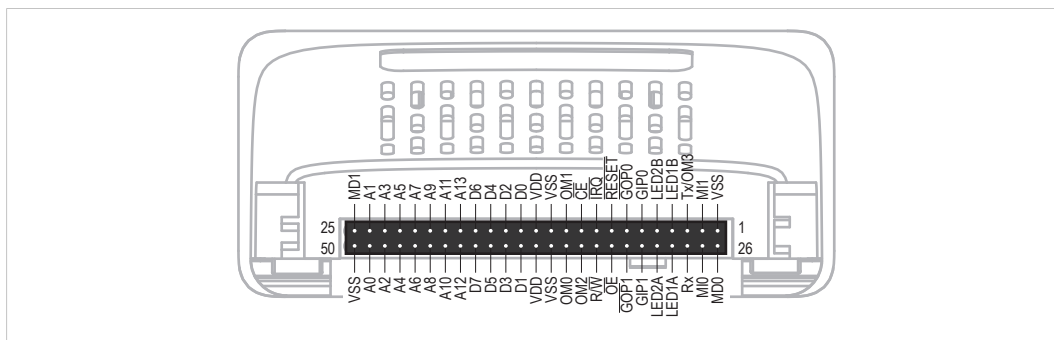


图 13

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)、 「Application Connector Pin Overview」

モジュールの識別 (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。

i

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	Active Anybus CompactCom 30
HIGH	LOW	Active Anybus CompactCom 40

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のです。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)*、「Settings/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 Interface/D8-D15 (Data Bus)」。

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)*、3.2.3、「LED Interface/D8-D15 (Data Bus)」。

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 $V_{IH}=V_{DD}+0.2\text{ V}$ 、40シリーズでは $V_{IH}=3.45\text{ V}$ と指定されています。論理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Anybus CompactCom40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)*、3.2.5、「RMII — Reduced Media-Independent Interface」。

F.3 ソフトウェア全般

F.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)*、セクション「Memory Map」

F.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シリーズモジュールは通常10-100 μ秒以内に応答します（ピンポンの間隔）。40シリーズでは通常、2 μ秒以内に応答します。

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。

F.3.3 スタートアップ時のAnybus CompactCom からホストアプリケーションへのリクエスト

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。

F.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。また、以下の「Control Register CTRL_AUX-bit」および「Status Register STAT_AUX-bit」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 154 GOP[0~1]/LED4[A~B], ページ 154

F.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 CompactCom30 用サンプルドライバ全体に、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)*、「Control Register」。

F.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からリリースされたAnybus CompactCom30用サンプルドライバ全体に、この違いが当てはまりません。

関連情報： *Anybus CompactCom 40 Software Design Guide (HMSI-216-125)*、「Status Register」。

F.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とき、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からリリースされたAnybus CompactCom30用サンプルドライバ全体に、この違いが当てはまりません。

関連情報： *Anybus CompactCom 40 Software Design Guide (HMSI-216-125)*、「Control Register」。

F.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からリリースされたAnybus CompactCom30用サンプルドライバ全体に、この違いが当てはまります。

F.4 ネットワーク固有 – EtherCAT

F.4.1 ネットワークコンフィグレーションオブジェクト (04h)

Device IDインスタンスのインスタンス番号は、3 (30シリーズ) から1 (40シリーズ) に変更されています。

F.4.2 EtherCATオブジェクト (F5h)

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
#2, Product Code	デフォルト: 0000 0034h	デフォルト: 0000 0036h	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。
#6, Manufacturer Device Name	デフォルト: 「Anybus-CC EtherCAT」	デフォルト: 「CompactCom 40 EtherCAT」	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。

F.4.3 ESIファイル (エンジニアリングツールが使用する設定ファイル)

30シリーズから40シリーズにマイグレーションする場合、新しく更新したESIファイルが必要です。HMS Industrial Networks ABが提供するESIファイルジェネレータを利用できます。下記をご覧ください。

ESIファイルジェネレータ

HMS Industrial Networks ABウェブサイトではESIファイルジェネレータを入手いただけます。ジェネレータは特定の実装合致した最新のESIファイルを作成します。ESIジェネレータは30シリーズおよび40シリーズの両方で動作します。

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

ESIファイルジェネレータは、30シリーズと40シリーズの以下の違いを含め最新のデータを作成します。

Product Code、Revision Number、Product Nameはモジュールの情報を反映して更新されます。注：これらの値はEtherCATオブジェクト（F5h）で変更することができ、ESIファイルの値はEtherCATオブジェクトの値と一致していなければなりません。

```
<Type ProductCode="#x00000036" RevisionNo="#x00020001">
  CompactCom 40 EtherCAT</Type>
```

最新仕様のEtherCAT state transition timeoutsがESIファイル内に記載されなければなりません。注：これらのタイムアウト値は、EtherCATオブジェクト（F5h）で変更でき、ESIファイルの値はEtherCATオブジェクトの値と一致していなければなりません。

```
<StateMachine>
  <Timeout>
    <PreopTimeout>1000</PreopTimeout>
    <SafeopOpTimeout>5000</SafeopOpTimeout>
    <BackToInitTimeout>1000</BackToInitTimeout>
    <BackToSafeopTimeout>200</BackToSafeopTimeout>
  </Timeout>
</StateMachine>
```

sync manager start addressesは、40シリーズでは変更されており、sync manager sizeはEtherCAT設定ツールで設定可能となりました。

```
<Sm MinSize="34" MaxSize="1486" DefaultSize="276" StartAddress="#x4000"
ControlByte="#x26" Enable="1">MBoxOut</Sm>
<Sm MinSize="34" MaxSize="1486" DefaultSize="276" StartAddress="#x4800"
ControlByte="#x22" Enable="1">MBoxIn</Sm>
<Sm StartAddress="#x2800" ControlByte="#x20" Enable="1">Inputs</Sm>
```

40シリーズは、File over EtherCAT (FoE)サポートしており、ESIファイルに反映されなければなりません。FoEがEtherCATホストオブジェクト内で無効になっている場合、このキーワードはESIファイルから削除されなければなりません。

```
<FoE/>
```

40シリーズはHMS製スレーブコントローラを使用しているため、以下の設定に沿ってEEPROMバイトサイズとSIIコンフィグレーションデータを変更しなければなりません。

```
<ByteSize>384</ByteSize>
<ConfigData>80360046F4010000000000000000</ConfigData>
```

40シリーズはboot strap stateをサポートしており、以下のキーワードを必要とします。

```
<BootStrap>0040000400480004</BootStrap>
```


G 著作権表示

印刷物の書式ルーチン

Copyright (C) 2002 Michael Ringgaard. 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、および以下の免責条項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、および下記の免責条項を、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、しかし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、および以下の免責条項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、および下記の免責条項を、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、しかし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なるかを問わず、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用FTPデーモンの一部です

FatFs - FAT ファイルシステムモジュールR0.09b (C)ChaN, 2013

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。

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.
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、および以下の免責条項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、および下記の免責条項を、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、しかし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。

Copyright 2013 jQuery Foundationおよび貢献者一同
<http://jquery.com/>

誰でも無償でこのソフトウェアと関連ドキュメンテーションファイル（以下「ソフトウェア」）のコピーを一本入手して、制約なくソフトウェアの使用・コピー・修正・融合・公表・配布・サブライセンス許諾を行ったり、ソフトウェアのコピーを販売したり、ソフトウェアの受領者にこれらの行為を認めたりすることが、以下の条件の下に認められます。

上記の著作権表示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、しかし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、ソフトウェア自体から、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、契約行為によるか、不正行為によるか、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および貢献者一同

誰でも無償でこのソフトウェアと関連ドキュメンテーションファイル（以下「ソフトウェア」）のコピーを一本入手して、制約なくソフトウェアの使用・コピー・修正・融合・公表・配布・サブライセンス許諾を行ったり、ソフトウェアのコピーを販売したり、ソフトウェアの受領者にこれらの行為を認めたりすることが、以下の条件の下に認められます。

上記の著作権表示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、しかし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、ソフトウェア自体から、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、契約行為によるか、不正行為によるか、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ライセンス

Copyright (c) 2012 Michael McLaughlin

誰でも無償でこのソフトウェアと関連ドキュメンテーションファイル（以下「ソフトウェア」）のコピーを一本入手して、制約なくソフトウェアの使用・コピー・修正・融合・公表・配布・サブライセンス許諾を行ったり、ソフトウェアのコピーを販売したり、ソフトウェアの受領者にこれらの行為を認めたりすることが、以下の条件の下に認められます。

上記の著作権表示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、しかし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、ソフトウェア自体から、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、契約行為によるか、不正行為によるか、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、および以下の免責条項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、および下記の免責条項を、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。

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って表示することはできません。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なければならず、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. この通知はどのようなソース配布においても削除したり改変したりすることはできません。

L. Peter Deutsch
ghost@aladdin.com

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、および以下の免責条項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、および下記の免責条項を、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、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、しかし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。

このページは意図的に空白になっています

