

Anybus[®] CompactCom 40 EtherCAT

Network Guide

Doc.Id. HMSI-27-321
Rev. 1.21



HALMSTAD • CHICAGO • KARLSRUHE • TOKYO • BEIJING • MILANO • MULHOUSE • COVENTRY • PUNE • COPENHAGEN

HMS Industrial Networks
Mailing address: Box 4126, 300 04 Halmstad, Sweden
Visiting address: Stationsgatan 37, Halmstad, Sweden

E-mail: info@hms-networks.com
Web: www.anybus.com

必ずお読みください

本ドキュメントは、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 40 に関する一般的な情報については、Anybus CompactCom 40 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。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。本製品のユーザーは、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。

責任の範囲

本マニュアルは細心の注意を払って作成されています。誤字や脱字があった場合は、HMS Industrial Networks AB にお知らせください。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、製品開発に継続的に取り組むという自社のポリシーに基づき、製品に変更を加える権利を留保します。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。本装置のユーザーは、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ぬものとします。

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の責任を負いません。本製品のかかる側面を直接または間接に使用したことによって生じる影響（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や図表は、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や出願中の特許が含まれます。

商標

Anybus® は、HMS Industrial Networks AB の登録商標です。その他の商標は、各所有者に帰属します。



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。

警告：	これはクラス A 製品です。ご家庭でお使いになる場合、電波障害を引き起こす場合があります。その場合は適切な措置をお取りください。
ESD に関する注意事項：	本製品では、ESD（静電気放電）による損傷を受けやすい部品が使用されています。ESD の管理手順に従わない場合、それらの部品が損傷するおそれがあります。本製品を扱う際は、静電気を管理するための予防措置を講じてください。この予防措置を怠った場合、本製品が損傷するおそれがあります。

Anybus CompactCom 40 EtherCAT Network Guide

Rev 1.21

Copyright© HMS Industrial Networks AB

Feb 2015 Doc.Id. HMSI-27-321

目次

前書き	本ドキュメントについて	
	関連ドキュメント	5
	ドキュメント更新履歴	5
	表記と用語	6
	サポート	6
第 1 章	Anybus CompactCom 40 EtherCAT について	
	概要	7
	機能	7
第 2 章	基本動作	
	概要	8
	ソフトウェアの要件	8
	EtherCAT スレーブインターフェース (ESI) ファイル	9
	デバイス Identity	9
	File Access over EtherCAT (FoE)	10
	EtherCAT の実装詳細	11
	概要	11
	EtherCAT の同期	11
	シンクマネージャー	11
	FMMU	12
	アドレッシングモード	12
	ウォッチドッグ機能	13
	CANopen over EtherCAT の実装詳細	14
	概要	14
	実装されているサービス	14
	データ交換	15
	アプリケーションデータ (ADI)	15
	プロセスデータ	15
	ネットワークのリセット処理	16
	ノードのリセット	16
	メーカーのパラメータをデフォルトに戻す	16
	設定ステーションエイリアス (ノードアドレス)	16
	Device ID	16
	モジュラーデバイスプロファイル	17

第 3 章	オブジェクトディクショナリ (CANopen over EtherCAT)	
	標準オブジェクト	18
	概要	18
	オブジェクトエントリ	19
	メーカー及び、プロファイル固有オブジェクト	23
	概要	23
	ネットワークデータの形式	23
	エラーコード	24
	オブジェクトエントリ	25
	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	26
第 4 章	Anybus モジュールオブジェクト	
	概要	29
	Anybus オブジェクト (01h)	30
	診断オブジェクト (02h)	31
	ネットワークオブジェクト (03h)	33
	ネットワークコンフィグレーションオブジェクト (04h)	35
第 5 章	ホストアプリケーションオブジェクト	
	概要	37
	アセンブリマッピングオブジェクト (EBh)	38
	SYNC オブジェクト (EEh)	39
	EtherCAT オブジェクト (F5h)	42
Appendix A	機能の分類	
	基本	47
	拡張	47
	高度	47
Appendix B	実装詳細	
	SUP ビットの定義	48
	Anybus ステートマシン	48
	アプリケーションウォッチドッグのタイムアウト処理	48

Appendix C技術仕様

前面図	49
保護接地（PE）に関する要件	50
電源	51
環境仕様	51
EMC への準拠	51

Appendix Dタイミングと性能

概要	52
内部タイミング	52
起動遅延	52
NW_INIT の処理	52
イベントベースの WrMsg ビジー時間	53
イベントベースのプロセスデータ遅延時間	53

Appendix E著作権表示

P. 本ドキュメントについて

より詳しい情報や各種ドキュメントは、HMS の Web サイト www.anybus.jp から入手いただけます。

P.1 関連ドキュメント

ドキュメント	作成者
Anybus CompactCom 40 Software Design Guide	HMS
Anybus CompactCom 40 Hardware Design Guide	HMS
Anybus CompactCom 40 Driver User Manual	HMS
IEC 61158-6	IEC
CiA ドラフト規格 301 v4.02	CAN in Automation

P.2 ドキュメント更新履歴

最近の変更に関する概要（1.20 ～ 1.21）

変更内容	ページ
ネットワークデータの形式に関する注 1 を更新	23
Anybus オブジェクトに Network time アトリビュートを追加	30
EtherCAT ホストオブジェクト（F5h）にアトリビュート 14 と 15 を追加	42

リビジョンリスト

リビジョン	日付	作成者	章	説明
1.00	2014/03/21	KeL	すべて	最初の正式リリース
1.01	2014/04/04	KeL		商標に EtherCAT ロゴを追加
1.10	2014/07/15	KeL	2, 3, 4, C, D, E	更新と追加
1.20	2015/01/19	KeL	2, 3, 4, 5, B, C, D	各種更新と訂正
1.21	2015/02/24	KaD	3, 4, 5	小規模な変更

P.3 表記と用語

本マニュアルでは以下の表記を使用します。

- 番号付きリストは手順を表します。
- 番号なしリストは情報を表します。手順ではありません。
- "Anybus" または "モジュール" は、Anybus CompactCom モジュールを表します。
- "ホスト" または "ホストアプリケーション" は、Anybus モジュールをホスト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。

P.4 サポート

お問い合わせとサポートに関する情報は、www.anybus.jp の各ページをご覧ください。

1. Anybus CompactCom 40 EtherCAT について

1.1 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通じて、EtherCAT 仕様への適合性が確認された通信を簡単に実現します。本モジュールの機能は、EtherCAT の規格に準拠しているあらゆる機器で利用することができます。これにより、種類を問わず、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が可能となります。

本製品は、Anybus CompactCom Hardware / Software Design Guide で定義されたアクティブモジュールが使用するあらゆるホストインターフェースに対応しています。そのため、EtherCAT の規格に準拠したあらゆる機器との間で完全なデータ通信が可能です。通常、ネットワーク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機能を使用する際に専用のソフトウェアが必要となる場合があります。

1.2 機能

- CANopen over EtherCAT (CoE)
- モジュラーデバイスプロファイルのサポート
- RJ45 コネクター
- 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 のサポート
- 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
- ディストリビュートクロックを使用した Sync0 機能のサポート
- 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 100 μ s 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

2. 基本動作

2.1 概要

2.1.1 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 ネットワークシステムの特性上、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、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、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。

Anybus CompactCom 40 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。

2.1.2 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。

HMS は、標準の ESI ファイルを提供します。このファイルは、新たな開発を行うときのベースとして使用できます。ただし、Anybus CompactCom コンセプトは柔軟であるため、モジュールの機能を変更した結果、このファイルに記述されている情報と矛盾する場合があります。その場合、マスターがこのファイルに記述された設定を期待していると、不具合が発生することがあります。そのような不具合は、例えばエンドユーザーが I/O パラメータを手動で変更するなどの方法で修正できる場合があります。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。

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。
- 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 ファイルに従うために、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。

2.1.3 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 001Bh (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) のサポートが実装された場合、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 のコンフォーマンス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。

- 42 ページの「EtherCAT オブジェクト (F5h)」

2.1.4 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ー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 hiff (HMS ファームウェアファイル) または nfo のファイルに関係しない FoE 要求はすべて、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に転送されます。詳細については、『Anybus CompactCom 40 Software Design Guide』を参照してください。

FoE によってダウンロードされたファームウェアファイルの更新が保留されている場合、拡張子 hiff のファイルを FoE によってアップロードできません。

2.2 EtherCAT の実装詳細

2.2.1 概要

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
- シンクマネージャー： 4
- RAM サイズ： 16 k バイト

下記も参照してください。

- 14 ページの「CANopen over EtherCAT の実装詳細」

2.2.2 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。

2.2.3 シンクマネージャー

本モジュールには 4 つのシンクマネージャーが用意されています。

- **シンクマネージャー 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。
- **シンクマネージャー 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。
- **シンクマネージャー 2**
RxPDO が格納されます（実際は、シンクマネージャー 2 にはリードプロセスデータが格納されます）。
- **シンクマネージャー 3**
TxPDO が格納されます（実際は、シンクマネージャー 3 にはライトプロセスデータが格納されます）。

2.2.4 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。

2.2.5 アドレッシングモード

EtherCAT のスレーブと通信する際に使用可能な様々なアドレッシングモードが用意されています。本モジュールは EtherCAT スレーブ機器のすべての機能を備えており、ポジションアドレッシング、ノードアドレッシング、論理アドレッシングをサポートしています。

2.2.6 ウォッチドッグ機能

標準のウォッチドッグ機能に加え、以下のウォッチドッグが実装されています。

- **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によりリードプロセス データが更新されなかった場合、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。

- 48 ページの「SUP ビットの定義」

2.3 CANopen over EtherCAT の実装詳細

2.3.1 概要

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。

- 15 ページの「データ交換」
- 18 ページの「オブジェクトディクショナリ (CANopen over EtherCAT)」

2.3.2 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 の高速ダウンロード	スレーブに最大 4 オクテットを書き込む
SDO の通常ダウンロード	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
SDO セグメントのダウンロード	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 の高速アップロード	スレーブから最大 4 オクテットを読み出す
SDO の通常アップロード	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
SDO セグメントのアップロード	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
SDO の転送中止	エラー発生時にサーバーがサービスを停止
OD リストの取得	利用可能なインデックスのリストを読み出す
オブジェクトに関する記述の取得	インデックスの詳細を読み出す
エントリに関する記述の取得	サブインデックスの詳細を読み出す
緊急事態	予期せぬ状態や診断イベントを報告する

2.4 データ交換

2.4.1 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。

- 23 ページの「メーカー及び、プロファイル固有オブジェクト」

2.4.2 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、プロセスデータオブジェクト (PDO) としてバス上で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に基づいて決定されるか、アプリケーションの実装方法に従います。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。

注: 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ことを推奨します。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。

- 18 ページの「標準オブジェクト」
- 23 ページの「メーカー及び、プロファイル固有オブジェクト」
- 38 ページの「アセンブリマッピングオブジェクト (EBh)」
- アプリケーションデータオブジェクト (『Anybus CompactCom 40 Software Design Guide』を参照)
- モジュラーデバイスオブジェクト (『Anybus CompactCom 40 Software Design Guide』を参照)

2.5 ネットワークのリセット処理

2.5.1 ノードのリセット

有効なファームウェアが FoE (File access over EtherCAT) によってダウンロードされた場合、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。

2.5.2 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" (メーカーのパラメータをデフォルトに戻す) 要求をネットワークから受信すると、CmdExt[1] が 01h 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 (工場出荷状態へのリセット)。

工場出荷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。それ以外の状態のときにリセットを実行すると、SDO アポートコード 08000020h (無効な状態) が生成されます。

下記も参照してください。

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

2.6 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。

Anybus CompactCom 40 EtherCAT スレーブモジュールは、ローカルで設定された設定ステーションエイリアスをサポートしません。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。

2.7 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に役立ちます¹。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。²

下記も参照してください。

- 35 ページの「ネットワークコンフィグレーションオブジェクト (04h)」

1. これはホットコネクトと呼ばれます。
2. 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 がデバイス ID に使用されていたことに注意してください。

2.8 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をサポートします。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しているモジュールでは、最大 63 個のスロット（スロット 0 のカプラーを含む）がサポートされます。ネットワークからアクセスできる ADI の最大数は 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』）
- 26 ページの「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」

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

3.1 標準オブジェクト

3.1.1 概要

DS301 通信プロファイルに従って、標準オブジェクトの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。

3.1.2 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	種類	アクセス	メモ
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)。EtherCAT オブジェクトを通じて管理できます。EtherCAT オブジェクトは、必要に応じてホストアプリケーションに実装できます。42 ページの「EtherCAT オブジェクト (F5h)」を参照。 注: 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。31 ページの「診断オブジェクト (02h)」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	メーカーの機器名	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します。これは、ホストアプリケーションで実装することもできます。42 ページの「EtherCAT オブジェクト (F5h)」を参照。
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	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します。これは、ホストアプリケーションで実装することもできます。42 ページの「EtherCAT オブジェクト (F5h)」を参照。
		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	
1600h - 1xxxh	Receive PDO mapping	00h	PDO にて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 ^a	マッピングされたオブジェクトの数 (0 ~ 254)。詳細は 21 ページの「PDO に対する ADI のマッピング」を参照。
		01h	マッピングされたオブジェクト #1	U32	RO/RW ^a	-
		02h	マッピングされたオブジェクト #2	U32	RO/RW ^a	-
		-
		NNh	マッピングされたオブジェクト #NN	U32	RO/RW ^a	-

インデックス	オブジェクト名	サブインデックス	説明	種類	アクセス	メモ
1A00h - 1xxxh	Transmit PDO mapping	00h	PDO にて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 ^a	マッピングされたオブジェクトの数 (0 ~ 254)。詳細は 21 ページの「PDO に対する ADI のマッピング」を参照。
		01h	マッピングされたオブジェクト #1	U32	RO/RW ^a	-
		02h	マッピングされたオブジェクト #2	U32	RO/RW ^a	-
		-
		NNh	マッピングされたオブジェクト #NN	U32	RO/RW ^a	-
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 ^b	
		01h - NNh	割り当てられた PDO	U16	RO/RW ^b	
1C13h	Sync Manager Tx PDO Assign	00h	割り当てられた PDO の数	U8	RO/RW ^c	
		01h - NNh	割り当てられた PDO	U16	RO/RW ^c	
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	同期モード	U16	RO/RW	00h : フリーラン 02h : DC Sync0 39 ページの「SYNC オブジェクト (EEh)」も参照してください。
		02h	サイクルタイム	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 を設定 : FREE_RUN をサポート。 ビット 2 を設定 : DC Sync0 をサポート。 ビット 5 を設定 : 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。 39 ページの「SYNC オブジェクト (EEh)」も参照してください。
		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

インデックス	オブジェクト名	サブインデックス	説明	種類	アクセス	メモ
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	同期モード	U16	RO/RW	00h : フリーラン 02h : DC Sync0 39 ページの「SYNC オブジェクト (EEh)」も参照してください。
		02h	サイクルタイム	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 を設定 : FREE_RUN をサポート。 ビット 2 を設定 : DC Sync0 をサポート。 ビット 5 を設定 : 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。 39 ページの「SYNC オブジェクト (EEh)」も参照してください。
		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。

- a. 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)。
- b. 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。動的な PDO マッピングを使用している場合は書き込むことができます。
- c. 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。

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。

モード	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
一般、静的マッピング	RO	1 ~ 6 ^a	1 ~ 254 ^b
一般、動的マッピング	RW	1 ~ 6 ^a	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)
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 ^c	RO/RW ^d	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)
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

モード	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
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)

- a. 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。
- b. 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。
- c. 詳細は 38 ページの「アセンブリマッピングオブジェクト (EBh)」を参照してください。
- d. 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。

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。

モード	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。

3.2 メーカー及び、プロファイル固有オブジェクト

3.2.1 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 アポートコードに変換されます。

重要：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。

3.2.2 ネットワークデータの形式

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 データタイプ	ネットワークデータタイプ
BOOL	UNSIGNED8
SINT8	INTEGER8
SINT16	INTEGER16
SINT32	INTEGER32
UINT8	UNSIGNED8
UINT16	UNSIGNED16
UINT32	UNSIGNED32
CHAR	VISIBLE_STRING
ENUM	UNSIGNED8 または ENUM
BITS8	BITARR8
BITS16	BITARR16
BITS32	BITARR32
OCTET	OCTET_STRING
SINT64	INTEGER64
UINT64	UNSIGNED64
FLOAT	REAL32
PAD0-16	NULL
BIT1 - BIT7	BIT1 - BIT7

注 1：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（異なるデータ型の要素を混在させることが可能）で表されます。これに対する例外は "CHAR"（常に VISIBLE_STRING で表される）と "OCTET"（常に OCTET_STRING で表される）です。

注 2：要素を 1 つだけ持つ ADI は単変数で表されます。ただし、"CHAR"（常に VISIBLE_STRING で表される）と "OCTET"（常に OCTET_STRING で表される）は例外です。

3.2.3 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)
Reserved	N/A	-
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。
Object Specific Error	0800 0000h	一般的なエラー

注：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) になります。

3.2.4 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 は配列ではなく単純変数で表されます。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	種類	アクセス	メモ
2001h	ADI 0001h	00h	エントリ数 (NNh)	U8	RO	-
		01h	ADI 値 (アトリビュート #5) 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。	-	-	
		02h		-	-	
		...		-	-	
		...		-	-	
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) 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。	-	-	
		02h		-	-	
		...		-	-	
		...		-	-	
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) 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。	-	-	
		02h		-	-	
		...		-	-	
		...		-	-	
NNh	-	-				
...
5FFFh	ADI 3FFFh	00h	エントリ数 (NNh)	U8	RO	-
		01h	ADI 値 (アトリビュート #5) 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。	-	-	
		02h		-	-	
		...		-	-	
		...		-	-	
NNh	-	-				

3.2.5 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	種類	アクセス	メモ
6000h - 6FFFh	Input data	任意	カプ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、27 ページの「ADI から SDO への変換」を参照。
7000h- 7FFFh	Output data	任意	カプ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、27 ページの「ADI から SDO への変換」を参照。
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 (カプラーは除く)。	任意	RW	詳細については、28 ページの「モジュール識別オブジェクト」を参照。
F000h	Modular Device Profile	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	概要	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、カプ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。

インデックス	オブジェクト名	サブインデックス	説明	種類	アクセス	メモ
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。 ^a
		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	
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	-

a. 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。

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	種類	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	-

モジュール	ADI	種類	SDO
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 SDO 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	種類	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 です。(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 シリーズ内のオブジェクトはモジュールごとに作成され、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 シリーズ内のオブジェクトはモジュールごとに作成され、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、21 ページの「PDO に対する ADI のマッピング」を参照してください。

4. Anybus モジュールオブジェクト

4.1 概要

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：

- 30 ページの「Anybus オブジェクト (01h)」
- 31 ページの「診断オブジェクト (02h)」
- 33 ページの「ネットワークオブジェクト (03h)」
- 35 ページの「ネットワークコンフィグレーションオブジェクト (04h)」
- 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

ネットワーク固有オブジェクト：

(なし)

4.2 Anybus オブジェクト (01h)

カテゴリ

基本

オブジェクトの説明

このオブジェクトには、共通の Anybus データがすべて組み込まれています。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。

サポートされているコマンド

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

オブジェクトアトリビュート (インスタンス #0)

(詳細については、『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	種類	値
1	Module type	Get	UINT16	0403h (Anybus CompactCom 40)
2 ~ 11	-	-	-	詳細については、『Anybus CompactCom 40 Software Design Guide』を参照。
12	LED colors (LED の色)	Get	構造 : 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、『Anybus CompactCom 40 Software Design Guide』を参照。
17	Virtual attributes	Get/Set		
18	Black list/White list	Get/Set		
19	Network time	Get	UINT64	ナノ秒で表された 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せん。

4.3 診断オブジェクト (02h)

カテゴリ

基本

オブジェクトの説明

このオブジェクトは、ホストアプリケーションのイベントと診断を処理する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。

サポートされているコマンド

オブジェクト : Get_Attribute
 Create
 Delete

インスタンス : Get_Attribute

オブジェクトアトリビュート

#	名前	アクセス	データタイプ	値
1 ~ 4	-	-	-	詳細については、『Anybus CompactCom 40 Software Design Guide』を参照。
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

基本

#	名前	アクセス	種類	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。

1. 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、新たなエントリが以下のように作成されます。

上位バイト (未使用)	(UINT32)	イベントコード	下位バイト 00h

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。

ビット番号	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 20h ~ 23h である。 または イベントコードが FFh で、なおかつ NW 固有情報の上位バイトが 20h ~ 23h である。
2	電圧	イベントコードが 30h ~ 33h である。 または イベントコードが FFh で、なおかつ NW 固有情報の上位バイトが 30h ~ 33h である。
3	温度	イベントコードが 40h ~ 42h である。 または イベントコードが FFh で、なおかつ NW 固有情報の上位バイトが 40h ~ 42h である。
4	通信エラー	イベントコードが 80h ~ 82h である。 または イベントコードが FFh で、なおかつ NW 固有情報の上位バイトが 80h ~ 82h である。 または Anybus の状態が 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 FFh である。 かつ NW 固有情報の上位バイトが FFh である。

3. 以下の情報を持つ EMCY オブジェクトが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	イベントコード	エラーレジスタ (1001h)	メーカー固有フィールド (未使用)				

注 1: 復旧不可能な重大なイベントを作成すると、Anybus が EXCEPTION 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。

注 2: 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。

4.4 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関する詳細は、『Anybus CompactCom 40 Software Design Guide』を参照してください。

サポートされているコマンド

オブジェクト : Get_Attribute
インスタンス : Get_Attribute
Set_Attribute
Get_Enum_String
Map_ADI_Write_Area
Map_ADI_Read_Area
Map_ADI_Write_Ext_Area
Map_ADI_Read_Ext_Area

オブジェクトアトリビュート (インスタンス #0)

(詳細については、『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	種類	値
1	Network type	Get	UINT16	0087h
2	Network type string	Get	CHAR の配列	"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。 ^a
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。 ^a
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。34 ページの「Exception information」を参照。
8 ~ 10	(予約)	-	-	詳細については、『Anybus CompactCom 40 Software Design Guide』を参照。

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

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)。

4.5 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。

サポートされているコマンド

オブジェクト : Get_Attribute
Reset

インスタンス : Get_Attribute
Set_Attribute

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	データタイプ	Value
1	名前	Get	CHAR の配列	"Network configuration"
2	リビジョン	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、"Device ID"）

16 ページの「Device ID」も参照してください。

拡張

変更は直ちに有効になります。

#	名前	アクセス	種類	説明
1	Name ^a	Get	CHAR の配列	"Device ID"
2	Data type	Get	UINT8	05h (= UINT16)
3	Number of elements	Get	UINT8	01h (要素数 1)
4	Descriptor	Get	UINT8	03h (読み出し / 書き込みアクセス)
5	Value	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	デバイス ID の設定値。この値は常にアトリビュート #5 の値と同じになります。

a. Multilingual（多言語）。36 ページの「多言語文字列」を参照。

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

リセット

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. ホストアプリケーションオブジェクト

5.1 概要

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

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）
- アプリケーションデータオブジェクト（『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）
- 38 ページの「アセンブリマッピングオブジェクト (EBh)」
- 39 ページの「SYNC オブジェクト (EEh)」
- モジュラーデバイスオブジェクト（『Anybus CompactCom 40 Software Design Guide』を参照）

ネットワーク固有オブジェクト：

- 42 ページの「EtherCAT オブジェクト (F5h)」

5.2 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	方向	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。

注：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。

下記も参照してください。

- 『Anybus CompactCom 40 Software Design Guide』の「アセンブリマッピングオブジェクト」
- 18 ページの「標準オブジェクト」の、アセンブリと PDO のマッピングに関する説明

5.3 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 レジスタで問題を通知します。41 ページの「アプリケーションステータスレジスタ」を参照してください。

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」

サポートされているコマンド

オブジェクト : Get_Attribute

インスタンス : Get_Attribute
Set_Attribute

オブジェクトアトリビュート (インスタンス #0)

(詳細については、『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 で以下のように表されます。

#	名前	アクセス	種類	デフォルト値	コメント
1	Cycle time	Get/Set	UINT32		アプリケーションのサイクルタイム (単位: ナノ秒) 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 ^a	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、出力計算およびコピー時間)
5	Input processing ^a	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、入力計算およびコピー時間)
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 Parameter および SM Input Parameter、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 Parameter および SM Input Parameter、同期の種類)

#	名前	アクセス	種類	デフォルト値	コメント
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。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 Parameter および SM Input Parameter、サポートされている同期の種類)

a. 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。

アプリケーションステータスレジスタ

アプリケーションがアプリケーションステータスレジスタ (H_APPSTATUS) 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。H_APPSTATUS は次の表のように変換されて、ALStatusCode レジスタに格納されます。

H_APPSTATUS	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	SYNCEERROR (001Ah)	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	DCSYNCEOERROR (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。

5.4 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。

注 1: 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。

注 2: 最終製品がコンフォーマンス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

#	名前	アクセス	データタイプ	値
1	Name	Get	CHAR の配列	"EtherCAT"
2	Revision	Get	UINT8	01h
3	Number of instances	Get	UINT8	01h
4	Highest instance no.	Get	UINT8	01h

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	種類	デフォルト値	コメント
1	Vendor ID	Get	UINT32	0000 001Bh	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))

拡張

#	名前	アクセス	種類	デフォルト値	コメント
2	Product Code	Get	UINT32	0000 0034h	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 Hardware Version (メーカーのハードウェアバージョン) Device Name	Get	CHAR の配列 (最大 32 バイト)	製品固有	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 (メーカーのハードウェアバージョン) Hardware Version	Get	CHAR の配列 (最大 32 バイト)	x.yy	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 Hardware Version (メーカーのハードウェアバージョン) Software Version	Get	CHAR の配列 (最大 32 バイト)	x.yy	オブジェクトエントリ 100Ah の値を指定します (Manufacturer Software Version)
9	ENUM ADIs	Get	UINT16 の配列	-	デフォルトでは、ENUM は EtherCAT 上の UNSIGNED8 に変換されます。このアトリビュートを実装すると、ENUM はバス上の ENUM にも変換されます。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	製品固有	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をアプリケーションで使用して、アセンブリマッピングオブジェクトに含まれるライト 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" アトリビュートの同じインデックスにあるインスタンスに対して使用する TxPDO マッピングオブジェクトの番号を格納します。 有効な値 : 1A00h - 1BFFh.

#	名前	アクセス	種類	デフォルト値	コメント
12	Read PD assembly instance translation	Get	UINT16 の配列	空	このアトリビュートをアプリケーションで使用して、アセンブリマッピングオブジェクトに含まれるリード PD インスタンスのデフォルトの RxPDO マッピングオブジェクトを変更できます。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" アトリビュートの同じインデックスにあるインスタンスに対して使用する RxPDO マッピングオブジェクトの番号を格納します。 有効な値 : 1600h - 17FFh.
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 ~ 1FFFh) 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 から成るパ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。 例を参照。「ADI 変換の例」
14	(予約)	-	-	-	(将来のために予約)

#	名前	アクセス	種類	デフォルト値	コメント
15	Object subindex translation	Get	構造体の配列 { UINT16 UINT16 UINT8 }	空	<p>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。</p> <p>現在このアトリビュートによって追加できるのは、オブジェクト 0x1C32 ("Output Sync Manager Parameter") および 0x1C33 ("Input Sync Manager Parameter") のサブインデックスのみです。</p> <p>このアトリビュートは、2つの UINT16 と1つの UINT8 から成るパ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 に対応付けるオブジェクト (2番目の UINT16) のサブインデックスを指定します。</p> <p>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。</p> <p>次を参照。「オブジェクトサブインデックス変換の例」</p>

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したいとします。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。

```
[
  {
    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したいとします。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。例えば、エントリ 0x1C32 : 32 に対して ADI 0xF0FD を作成し、エントリ 0x1C33 : 32 に対して ADI 0xF0FE を作成します。さらに、Object subindex 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。

```
[
  {
    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。

A. 機能の分類

Anybus CompactCom とアプリケーションのオブジェクトやアトリビュート、サービスは、3つのカテゴリ、すなわち、基本機能、高度な機能、拡張機能に分類されます。

A.1 基本

このカテゴリには、必ず実装または使用しなければならないオブジェクトやアトリビュート、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す。産業用ネットワークの基本機能が使用されます。

製品の認定を可能とする追加オブジェクトなども、このカテゴリに属します。

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。ネットワークとの間の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。

A.3 高度

このカテゴリのオブジェクトやアトリビュート、サービスを使用すると、特殊な機能やあまり使用されない機能が利用できるようになります。提供されているほとんどのネットワーク機能が有効となり、利用可能になります。通常、産業用ネットワークの仕様の確認が必要となります。

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監視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、マスターとの通信途絶を検出できます。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せん。

注：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

以下の表に、Anybus のステートマシンと EtherCAT のネットワーク状態との対応を示しています。

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 アプリケーションウォッチドッグのタイムアウト処理

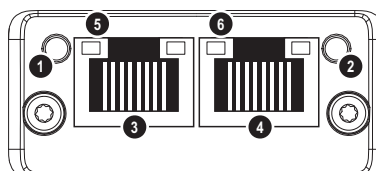
アプリケーションウォッチドッグがタイムアウトした場合、Anybus CompactCom 40 EtherCAT モジュールは EXCEPTION 状態になります。

C. 技術仕様

C.1 前面図

イーサネットコネクタ

#	項目
1	RUN LED ^a
2	ERROR LED ^a
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)



a. これらの 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。

RUN LED

この LED は、EtherCAT 通信の状態を表します。

LED の状態	意味	説明
オフ	INIT	EtherCAT 機器は "INIT" 状態にあります (または電源がオフです)。
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。
ちらつき	BOOT	EtherCAT 機器は "BOOT" 状態にあります。
赤 ^a	(致命的な事象)	-

a. RUN LED または ERR LED が赤く点灯した場合、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。

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 (または電源がオフです)。
赤点滅	無効な設定	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	アプリケーションのウォッチドッグのタイムアウト	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。
赤 ^a	アプリケーションコントローラの故障	Anybus モジュールが EXCEPTION 状態にあります。
ちらつき	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。

a. RUN LED または ERR LED が赤く点灯した場合、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。

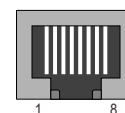
リンク / アクティビティ

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（または電源がオフになっています）。
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。
緑点滅	リンク検出。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。

イーサネットコネクタ（RJ45）

ピン	信号	メモ
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.2 保護接地（PE）に関する要件

適切な EMC 動作を保証するには、『Anybus CompactCom 40 Hardware Design Guide』に記述されている PE パッド / PE メカニズムに従って、本モジュールを適切に保護接地する必要があります。

PE に関する要件が満たされていない場合、HMS Industrial Networks は適切な EMC 動作を保証しません。

C.3 電源

電源電圧

本モジュールを使用するには、『Anybus CompactCom 40 Hardware Design Guide』で指定された 3.3 V の安定化電源が必要です。

消費電力

Anybus CompactCom EtherCAT は、クラス B モジュールの要件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¹。

注：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、『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。

C.4 環境仕様

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

C.5 EMC への準拠

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

-
1. HMS Industrial Networks は、製品開発に継続的に取り組むという HMS のポリシーに従って、本製品の消費電力に関する厳密な要件を予告なく変更する権利を留保します。ただし、いかなる場合であっても、Anybus CompactCom 40 EtherCAT はクラス B モジュールであることを維持します。

D. タイミングと性能

D.1 概要

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
起動遅延	T1, T2	52
NW_INIT の処理	T100	52
イベントベースの WrMsg ビジー時間	T103	53
イベントベースのプロセスデータ遅延時間	T101, T102	53

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測定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 (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 / 処理可能となった (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。	32 ^a
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

a. または、ネットワーク固有の最大数がそれより小さい場合は、その最大数。

パラメータ	説明	通信	最大	単位
T100	NW_INIT の処理	イベントベースのすべてのモード	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

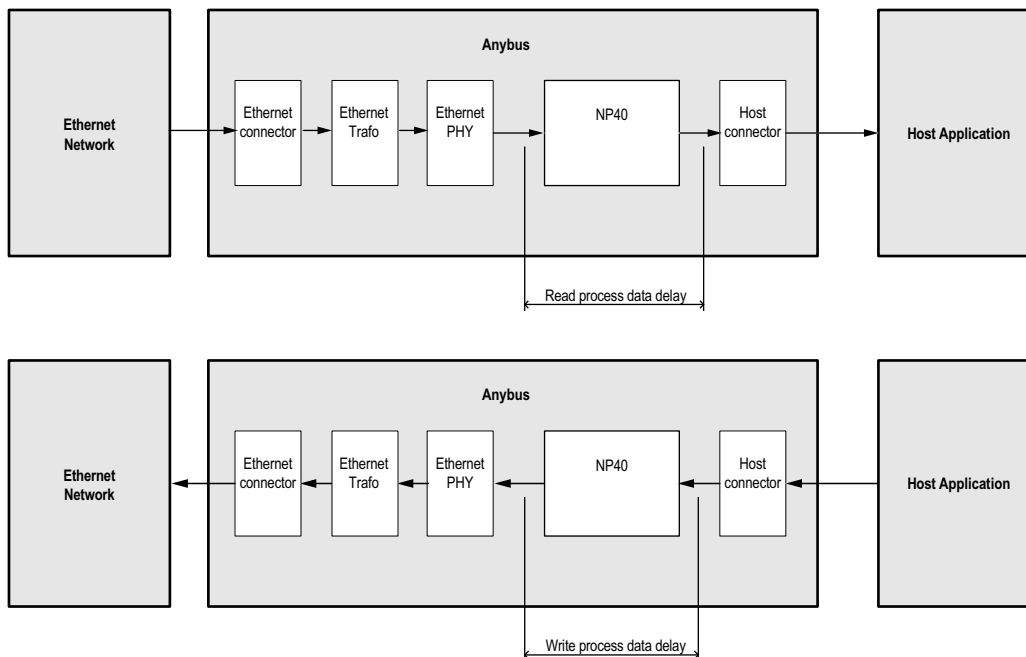
"リードプロセスデータ遅延時間" は、ネットワークフレームの最後のビットがネットワークインターフェースによって受信されてから、アプリケーションに対して RDPDI 割り込みがアサートされるまでの時間として定義されます。

"ライトプロセスデータ遅延時間" は、アプリケーションがライトプロセスデータバッファを交換してから、新規プロセスデータフレームの最初のビットがネットワークに送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 (代表値)	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



E. 著作権表示

Format - lightweight string formatting library.

Copyright (C) 2010-2013, Neil Johnson

All rights reserved.

ソースおよびバイナリ形式での再配布および使用は、変更の有無にかかわらず、以下の条件を満たす場合に許可されます。

* ソースコードを再配布する場合は、上記の著作権表示、本条件リスト、および以下の免責表示を含める必要があります。

* バイナリ形式で再配布する場合は、上記の著作権表示、本条件リスト、および下記の免責表示を、配布に際して提供する関連文書 および資料に記載する必要があります。

* 書面による事前の許可なしに、著作権保持者または 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、何の保証もなされません。いかなる場合においても、著作権所有者または貢献者は、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。