

# Software Design Guide

# Anybus<sup>®</sup>-CompactCom

Doc.Id. JCM-1201-008  
Rev. 2.03



---

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](mailto:info@hms-networks.com)  
Web: [www.anybus.com](http://www.anybus.com)

---

## 重要なユーザ情報

このマニュアルは、Active Anybus-CompactCom 通信モジュールで使用されるソフトウェア・インターフェースについて十分にご理解いただくことを意図しています。このマニュアルにはパッシブ・モジュールについては含まれておらず、またネットワーク特有の機能についても含まれていません。これらの情報は、別の文書（Network Interface Appendix）として提供されています。

このマニュアルの読者は、ハイ・レベルおよびロー・レベルのソフトウェア設計および一般的な通信システムに精通していることを前提としています。

### 責任

このマニュアルはあらゆる点を考慮して作成しています。不正確な記述や記載漏れがあった場合には、HMS Industrial Networks AB までご連絡ください。このマニュアルに含まれるデータや説明には拘束力はありません。HMS Industrial Networks AB は、継続的な製品開発を旨とする当社のポリシーに則って、弊社の製品を改良する権利を留保します。このマニュアルに含まれる情報は予告なく変更される場合があります。ただし、HMS Industrial Networks AB は変更に関して義務を負うものではありません。HMS Industrial Networks AB はこのマニュアルに現れるあらゆるエラーに対して責任を負いません。

この製品には多くのアプリケーションがあります。この装置の使用責任者は、アプリケーションが該当する法律、規則、規定、および規格を含む全ての性能および安全要求事項を満たしており、これを確認するために全ての必要な手順がとられたことを保証する必要があります。

HMS Industrial Networks AB は、いかなる状況においても、文書化されていない機能の使用、タイミング、またはこの製品の文書化された範囲外で見つかった機能面での副次的な影響によって発生する可能性がある問題に対する義務または責任を負いません。製品のこのような側面の直接的または間接的な使用によって発生する結果は不明確であり、互換性の問題や安定性の問題などを含む可能性があります。

このマニュアルに含まれる例および図表は、説明のためにのみ記載されています。特定の実装には多くの要素や要件が関連しているため、HMS Industrial Networks AB は、これらの例や図表に基づいた実際の使用に対する責任を負いません。

### 知的財産権

HMS Industrial Networks AB は、このマニュアルに記載された製品に組み入れられた技術に関する知的財産権を所有します。これらの知的財産権には、米国およびその他の国での特許および出願中の特許が含まれる可能性があります。

### 商標について

Anybus<sup>®</sup> は、HMS Industrial Networks AB の登録商標です。その他の全ての商標は、各所有者の資産です。

<b>警告：</b>	これはクラス A 製品です。国内の環境では、この製品は無線妨害を発生させる可能性があります。この場合、ユーザは適切な対策をとる必要があります。
------------	---

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

<b>目次</b>	
はじめに	このマニュアルについて
	関連マニュアル.....1
	マニュアル更新履歴.....1
	慣例と用語集.....2
	サポート.....3
<b>第1章</b>	<b>Anybus-CompactCom について</b>
	概要.....4
	特長.....4
<b>第2章</b>	<b>ソフトウェアについて</b>
	背景.....5
	オブジェクト・モデル.....6
	基本.....6
	アドレス指定方式.....6
	オブジェクトのカテゴリ.....6
	標準オブジェクトの実装.....7
	ネットワーク・データの交換.....8
	診断.....10
	多言語によるサポート.....10
<b>第3章</b>	<b>ホスト通信レイヤ</b>
	概要.....11
	基本原理.....11
	テレグラムの内容.....11
	ハンドシェーク・レジスタ.....12
	制御レジスタ (リード/ライト).....12
	ステータス・レジスタ (リード・オンリー).....12
	監視ビット (SUP).....13
	予備ビット (STAT_AUX、CTRL_AUX).....13
	プロセス・データ・サブフィールド.....14
	概要.....14
	プロセス・データのマッピング.....14
	変更データ表示.....15
	Anybus ウォッチドッグ.....16
	アプリケーション・
	ウォッチドッグ.....16
	シリアル・ホスト・テレグラム.....17
	概要.....17
	シリアル・テレグラム・フレーム.....17
	メッセージのフラグメンテーション.....18
	送信エラー.....19
	パラレル・ホスト・テレグラム.....21
	概要.....21
	メモリ・マップ.....21

	<i>パラレル・テレグラム処理</i> .....	22
<b>第4章</b>	<b>Anybus ステート・マシン</b>	
	概要 .....	23
	状態に依存する動作 .....	24
<b>第5章</b>	<b>オブジェクト・メッセージング</b>	
	概要 .....	25
	<i>基本原理</i> .....	25
	<i>送信元ID</i> .....	25
	<i>エラーの処理</i> .....	25
	メッセージのレイアウト .....	26
	データ・フォーマット .....	27
	<i>使用できるデータ・タイプ</i> .....	27
	<i>Char の配列の処理 (文字列)</i> .....	27
	フロー制御.....	28
	コマンドの仕様.....	29
	概要 .....	29
	<i>コマンド・コード</i> .....	30
	<i>エラー・コード</i> .....	30
	<i>Get_Attribute</i> .....	31
	<i>Set_Attribute</i> .....	31
	<i>Create</i> .....	32
	<i>Delete</i> .....	32
	<i>Reset</i> .....	33
	<i>Get_Enum_String</i> .....	33
	<i>Get_Indexed_Attribute</i> .....	34
	<i>Set_Indexed_Attribute</i> .....	34

<b>第 6 章</b>	<b>初期化と起動</b>	
	概要 .....	35
	最初のハンドシェーク .....	35
	Anybus のセットアップ ('SETUP' 状態) .....	36
	ネットワークの初期化 ('NW_INIT' 状態) .....	37
<b>第 7 章</b>	<b>Anybus モジュール・オブジェクト</b>	
	概要 .....	38
	オブジェクト改定 .....	38
	Anybus オブジェクト (01h) .....	39
	診断オブジェクト (02h) .....	43
	ネットワーク・オブジェクト (03h) .....	46
	ネットワーク・コンフィグレーション・オブジェクト (04h) .....	50
<b>第 8 章</b>	<b>ホスト・アプリケーション・オブジェクト</b>	
	概要 .....	52
	実装のガイドライン .....	53
	アプリケーション・データ・オブジェクト (FEh) .....	54
	コマンドの詳細: <i>Get_Profile_Instance_Numbers</i> .....	58
	コマンドの詳細: <i>Get_ADI_Info</i> .....	59
	コマンドの詳細: <i>Remap_ADI_Write_Area</i> .....	60
	コマンドの詳細: <i>Remap_ADI_Read_Area</i> .....	62
	アプリケーション・オブジェクト (FFh) .....	63
<b>アペンディックス A</b>	<b>機能のカテゴリ化</b>	
	基本 .....	68
	拡張 .....	68
	高度 .....	68
<b>アペンディックス B</b>	<b>ネットワーク比較チャート</b>	
<b>アペンディックス C</b>	<b>タイミングと性能</b>	
	概要 .....	71
	内部タイミング .....	72
	起動の遅延 .....	72
	NW_INIT 遅延 .....	72
	Anybus のレスポンス時間 .....	73
	概要 .....	73
	テレグラム遅延 .....	73
	コマンド遅延 .....	74
	プロセス・データ .....	75
	概要 .....	75
	Anybus リード・プロセス・データ遅延 ( <i>Anybus 遅延</i> ) .....	75
	Anybus ライト・プロセス・データ遅延 ( <i>Anybus 遅延</i> ) .....	76
	ネットワーク・システム・リード・プロセス・データ遅延 (ネットワーク・システム遅延) .....	76

---

	ネットワーク・システム・ライト・プロセス・データ遅延 (ネットワーク・システム遅延).....	76
<b>アペンディックス D</b>	<b>プロセス・データのランタイム再配置</b>	
	パラレル・モード.....	77
	リード・プロセス・データ.....	77
	ライト・プロセス・データ.....	78
	シリアル・モード.....	79
	リード・プロセス・データ.....	79
	ライト・プロセス・データ.....	80
	例 : Remap_ADI_Write_Area.....	81
<b>アペンディックス E</b>	<b>CRC の計算</b>	
	概要.....	82
	例.....	83

## P. このマニュアルについて

詳細な情報や資料などについては、HMS のウェブサイト（[www.anybus.com](http://www.anybus.com)）を参照してください。

### P.1 関連マニュアル

マニュアル名	作成者
ABCC Hardware Design Guide	HMS
ABCC Driver User Guides	HMS
ABCC Network Interface Appendix（各ネットワーク・システムに関する個別の文書）	HMS
ABCC Drive Profile Design Appendices（各ネットワーク・システムに関する個別の文書）	HMS

### P.2 マニュアル更新履歴

#### 最新の更新（2.02 ...2.03）

変更内容	ページ
サポート情報の更新	3
実装ガイドラインの追加	53
エラー・コード 06h の説明の明確化	30
メモリ・マップ内の制御レジスタのリード/ライトの変更	21
ネットワーク比較チャートの CompoNet の値の修正	70

#### 改定版リスト

改定番号	改定日	作成者	章	説明
1.00	2005-09-05	PeP	全て	最初の公式バージョン
1.01	2005-09-21	PeP	5、7、8	微調整
1.10	2007-01-28	PeP	全て	メジャー・アップデート
1.11	2007-07-09	PeP	3、7	マイナー・アップデート
1.12	2008-01-10	PeP	3、5、8、C	マイナー・アップデート
1.13	2008-10-23	HeS	3、5、7、A	マイナー・アップデート
1.20	2008-12-17	HeS	3、7、8	メジャー・アップデート
1.21	2009-04-30	PeP	-	マイナー・アップデート
1.23	2009-06-29	PeP	-	マイナー・アップデート
1.24	2009-11-20	KeL	3、4、5、7、 8、A、C	マイナー・アップデート
2.00	2010-04-13	KeL	3、7、8	マイナー・アップデート、オブジェクトのカテゴリ化
2.01	2010-08-20	KeL	7	マイナー・アップデート
2.02	2010-09-17	KeL	B	マイナー修正
2.03	2011-01-14	KeL		マイナー修正および更新

## P. 3 慣例と用語集

このマニュアルでは下記の慣例を使用しています。

- 番号が付いたリストは、連続した手順を示します。
- 黒丸が付いたリストは、手順ではなく情報を示します。
- 'Anybus' または 'モジュール' という用語は、Anybus-CompactCom モジュールを意味します。
- 'ホスト' または 'ホスト・アプリケーション' という用語は、Anybus-CompactCom のホストとして機能する装置を意味します。
- 16 進値は NNNNh または 0xNNNN というフォーマットでライトされます。NNNN は 16 進法の値です。
- 特に指示がないかぎり、Intel バイト・オーダーが想定されています。
- オブジェクト・インスタンスはインスタンス #0 と同じです。
- オブジェクト・アトリビュートはオブジェクト・インスタンスにあります。
- 'Anybus の実装' および 'Anybus バージョン' という用語は、一般的に Anybus モジュールでの実装（ネットワーク・タイプや内部ファームウェアの変更）を意味します。
- 任意であると明確に記載されていないかぎり、必須とみなすものとします。
- ライトする際には、'予約' と宣言されたフィールドはゼロに設定するものとします。
- リードする際には、'予約' と宣言されたフィールド・ビットは無視するものとします。
- '予約' と宣言されたフィールドは、記載されていない目的のために使用してはなりません。
- 1 バイトは常に 8 ビットで構成されています。



## P. 4 サポート

Sales		Support	
<b>HMS Sweden (Head Office)</b>			
E-mail:	sales@hms.se	E-mail:	support@hms-networks.com
Phone:	+46 (0) 35 - 17 29 56	Phone:	+46 (0) 35 - 17 29 20
Fax:	+46 (0) 35 - 17 29 09	Fax:	+46 (0) 35 - 17 29 09
Online:	www.anybus.com	Online:	www.anybus.com
<b>HMS North America</b>			
E-mail:	us-sales@hms-networks.com	E-mail:	us-support@hms-networks.com
Phone:	+1-312 - 829 - 0601	Phone:	+1-312-829-0601
Toll Free:	+1-888-8-Anybus	Toll Free:	+1-888-8-Anybus
Fax:	+1-312-629-2869	Fax:	+1-312-629-2869
Online:	www.anybus.com	Online:	www.anybus.com
<b>HMS Germany</b>			
E-mail:	ge-sales@hms-networks.com	E-mail:	ge-support@hms-networks.com
Phone:	+49 (0) 721-96472-0	Phone:	+49 (0) 721-96472-0
Fax:	+49 (0) 721-96472-10	Fax:	+49 (0) 721-96472-10
Online:	www.anybus.de	Online:	www.anybus.de
<b>HMS Japan</b>			
E-mail:	jp-sales@hms-networks.com	E-mail:	jp-support@hms-networks.com
Phone:	+81 (0) 45-478-5340	Phone:	+81 (0) 45-478-5340
Fax:	+81 (0) 45-476-0315	Fax:	+81 (0) 45-476-0315
Online:	www.anybus.jp	Online:	www.anybus.jp
<b>HMS China</b>			
E-mail:	cn-sales@hms-networks.com	E-mail:	cn-support@hms-networks.com
Phone:	+86 (0) 10-8532-3183	Phone:	+86 (0) 10-8532-3023
Fax:	+86 (0) 10-8532-3209	Fax:	+86 (0) 10-8532-3209
Online:	www.anybus.cn	Online:	www.anybus.cn
<b>HMS Italy</b>			
E-mail:	it-sales@hms-networks.com	E-mail:	it-support@hms-networks.com
Phone:	+39 039 59662 27	Phone:	+39 039 59662 27
Fax:	+39 039 59662 31	Fax:	+39 039 59662 31
Online:	www.anybus.it	Online:	www.anybus.it
<b>HMS France</b>			
E-mail:	fr-sales@hms-networks.com	E-mail:	fr-support@hms-networks.com
Phone:	+33 (0) 3 68 368 034	Phone:	+33 (0) 3 68 368 033
Fax:	+33 (0) 3 68 368 031	Fax:	+33 (0) 3 68 368 031
Online:	www.anybus.fr	Online:	www.anybus.fr
<b>HMS UK &amp; Eire</b>			
E-mail:	uk-sales@anybus.co.uk	E-mail:	support@hms-networks.com
Phone:	+44 (0) 1926 405599	Phone:	+46 (0) 35 - 17 29 20
Fax:	+44 (0) 1926 405522	Fax:	+46 (0) 35 - 17 29 09
Online:	www.anybus.co.uk	Online:	www.anybus.com
<b>HMS Denmark</b>			
E-mail:	info@anybus.dk	E-mail:	support@hms-networks.com
Phone:	+45 (0) 22 30 08 01	Phone:	+46 (0) 35 - 17 29 20
Fax:	+46 (0) 35 17 29 09	Fax:	+46 (0) 35 - 17 29 09
Online:	www.anybus.com	Online:	www.anybus.com
<b>HMS India</b>			
E-mail:	in-sales@anybus.com	E-mail:	in-support@hms-networks.com
Phone:	+91 (0) 20 40111201	Phone:	+46 (0) 35 - 17 29 20
Fax:	+91 (0) 20 40111105	Fax:	+46 (0) 35 - 17 29 09
Online:	www.anybus.com	Online:	www.anybus.com

# 1. Anybus-CompactCom について

## 1.1 概要

Anybus-CompactCom ネットワーク通信モジュールは、産業用フィールド・デバイスの高性能かつロー・コストの通信ソリューションです。一般的なアプリケーションは、周波数インバータ、PLC、HMI、表示装置、計測器、スケール、ロボット工学、およびインテリジェントな測定装置です。

Anybus-CompactCom ソフトウェア・インターフェースは、ネットワーク・プロトコルから独立しており、ホスト・アプリケーションが、機能を失うことなく同じソフトウェア・ドライバを使用して全ての主要なネットワークング・システムをサポートできるように設計されています。

拡張のためのフレキシビリティと余地を提供するために、ホスト・アプリケーションと Anybus モジュールの間にオブジェクト指向のアドレス指定方式が使用されています。ホスト通信プロトコルによって、Anybus モジュールは、メモリがマッピングされたデータではなく、明示的なオブジェクト要求を使用してホスト・アプリケーションから情報を直接取得できるため、非常にハイ・レベルな統合が可能になります。

HMS は、開発プロセスをスピードアップするために自由に使用できるフリー・ソース・レベル (C 言語) のソフトウェア・ドライバを供給します。ドライバは Anybus モジュールとホスト・アプリケーションの間で "接着剤" として作用します。ホスト・アプリケーション・ソフトウェアからロー・レベルのホスト・インターフェース通信を分離し、共通の Anybus 関連のタスクに使いやすい関数呼び出しを提供します。

ドライバは OS から完全に独立しており、必要に応じて、オペレーティング・システムがなくても使用できます。詳細については、"ABCC Driver User Guide" を参照してください。

**注意:** このマニュアルには、Passive Anybus-CompactCom モジュールについては含まれていません。詳細については、これらの製品の個別の Network Interface Appendix を参照してください。

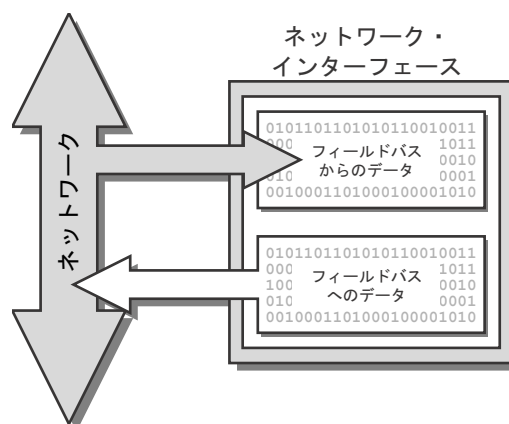
## 1.2 特長

- オブジェクト指向の技術
- ネットワーク・プロトコルに依存しない
- 多言語によるサポート
- ハイ・レベルな統合
- サイクリックおよびアサイクリック・データ
- 高度なネットワーク特有の機能に対する任意のサポート
- シリアルまたはパラレル操作
- 割り込み操作 (任意)
- フリー・ソース・レベル (C 言語) のソフトウェア・ドライバが使用可能

## 2. ソフトウェアについて

### 2.1 背景

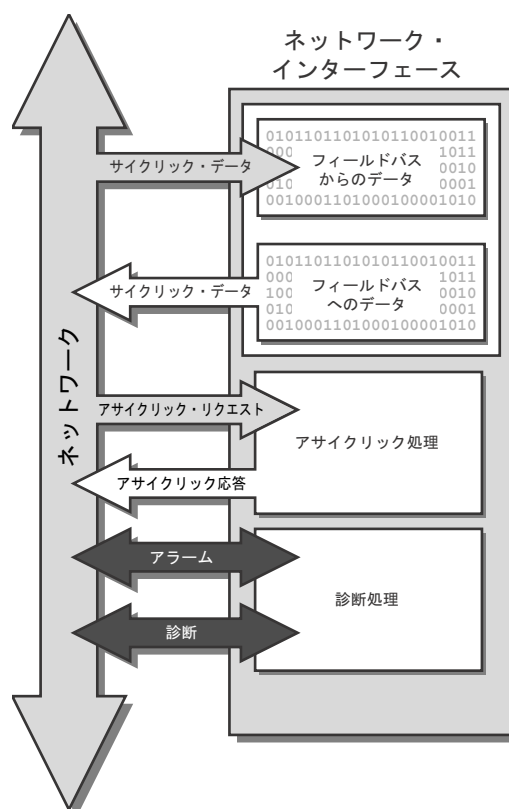
産業用ネットワーク・インターフェースの主な機能は、ネットワーク上の他のデバイスと情報を交換することです。従来これはほとんどの場合において、サイクリック I/O を交換し、2つのメモリ・バッファを介してホスト・デバイスで使用できるようにすることでした。



より高いレベルのネットワーク機能への要求が高まるにつれて、ネットワーク・インターフェースの一般的な役割は発展し、アサイクリック・データの管理、アラームの処理、診断などを含むようになりました。

ほとんどの場合、その実装方法は、ネットワーク・システムによって根本的に異なります。これは、各ネットワークに対する専用ソフトウェアのサポートを実装せずに、この新しい機能をサポートして実際に活用するということが、不可能ではないにしても、ますます複雑になってきていることを意味します。

最新のオブジェクト指向の技術を利用することによって、Anybus-CompactCom は、各ネットワークに対してほとんどのネットワーク・システムをサポートする簡単かつ効果的な方法を提供します。高度なネットワーク機能を活用でき、個別のソフトウェア・ドライバをライトする必要はありません。アサイクリック・リクエストは同様に交換され、専用オブジェクトは各ネットワーク標準に応じた診断およびアラーム処理を提供します。

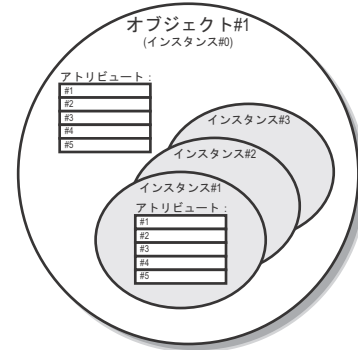


## 2.2 オブジェクト・モデル

### 2.2.1 基本

ホスト・アプリケーションと Anybus モジュールの両方にフレキシブルかつ論理的なアドレス指定方式を提供するために、ソフトウェア・インターフェースはオブジェクト指向で構成されています。このアプローチは最初は混乱するようと思われる可能性がありますが、情報をカテゴリ化してアドレス指定する方法にすぎません。

関連情報およびサービスは、'オブジェクト' と呼ばれるエンティティに分類されます。各オブジェクトは 'インスタンス' と呼ばれるサブエンティティを保持でき、インスタンスには 'アトリビュート' と呼ばれるいくつかのフィールドが含まれる場合があります。アトリビュートは、一般的にオブジェクトに関連付けられた情報または設定を表します。オブジェクトによっては、インスタンスは静的であるか、ランタイム中に動的に作成されます。



### 2.2.2 アドレス指定方式

オブジェクトとその内容には、オブジェクト・メッセージングを使用してアクセスします。各オブジェクト・メッセージには、メッセージに関連付けられたデータまたは設定の位置を指定するオブジェクト番号、インスタンス、およびアトリビュートがタグ付けされます。

#### 例:

モジュールは、Anybus モジュール自体に関連する共通設定を分類する 'Anybus オブジェクト' と呼ばれるオブジェクトを特徴としています。

このオブジェクトでは、インスタンス#1に 'Firmware version' と呼ばれるアトリビュート (アトリビュート #2) が含まれます。モジュールのファームウェア改定版を取得するために、ホストは単に 'get' 要求をオブジェクト #1 (Anybus オブジェクト)、インスタンス 1、アトリビュート 2 (Firmware version) に発行します。

---

**重要:** このアドレス指定方式は両方向に適用されます。Anybus モジュールと同様に、ホスト・アプリケーションは着信オブジェクト要求を解釈できる必要があります、これらを適切なソフトウェア・ルーティンにルーティングする必要があります。

### 2.2.3 オブジェクトのカテゴリ

その物理的な位置に基づいて、オブジェクトは2つの個別のカテゴリに分類されます。

- Anybus モジュール・オブジェクト**  
 これらのオブジェクトは Anybus ファームウェアの一部で、一般的にはモジュールの挙動およびネットワーク上での動作を制御します。
- ホスト・アプリケーション・オブジェクト**  
 これらのオブジェクトはホスト・アプリケーション・ファームウェアにあり、Anybus モジュールによってアクセスできます。これは、ホスト・アプリケーションが着信オブジェクト要求の適切な処理を実装する必要があることを意味します。

## 2.2.4 標準オブジェクトの実装

標準オブジェクトの実装は、全ての主要なネットワーク・システムのニーズに応じるように設計されています。これは、ネットワークのタイプに関係なく十分な機能性を得るためには、通常これらのオブジェクトに対するサポートを実装するだけで十分であることを意味します。

オプションとして、ネットワーク特有のオブジェクトのサポートは、ネットワーク特有の高度な機能へのアクセスを可能にするために実装できます。これらのオブジェクトについては、各 Network Interface Appendix で個別に説明しています。

### Anybus モジュール・オブジェクト

標準の Anybus-CompactCom ファームウェアでは、次のオブジェクトが実装されています。

- Anybus オブジェクト
- 診断オブジェクト
- ネットワーク・オブジェクト
- ネットワーク・コンフィグレーション・オブジェクト
- ネットワーク特有のオブジェクト

**注意：**これらのオブジェクトに正確にどの程度サポートを実装する必要があるかについては、ホスト・アプリケーションの要件によって異なります。下記も参照してください。

- 38 ページの“Anybus モジュール・オブジェクト”

### ホスト・アプリケーション・オブジェクト

ホスト・アプリケーションでは、次のオブジェクトを実装する必要があります。

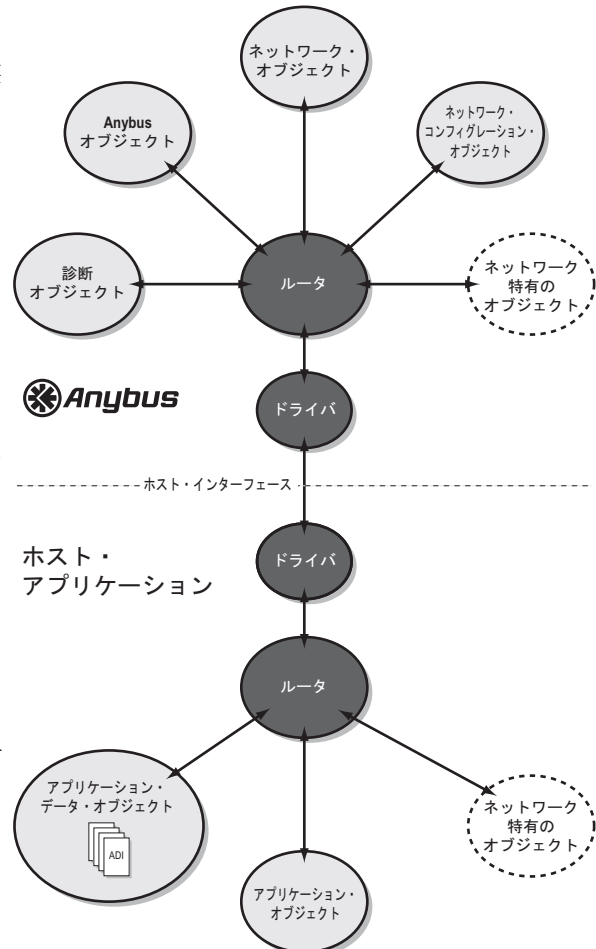
- アプリケーション・データ・オブジェクト (必須)
- アプリケーション・オブジェクト (任意)
- ネットワーク特有のオブジェクト (任意)

**注意：**アプリケーション・データ・オブジェクトを実装する必要があります。

ただし、正確な実装は、ホスト・アプリケーションの要件に大きく依存します。アプリケーション・オブジェクトの実装は任意ですが、特に推奨されています。

下記も参照してください。

- 52 ページの“ホスト・アプリケーション・オブジェクト”



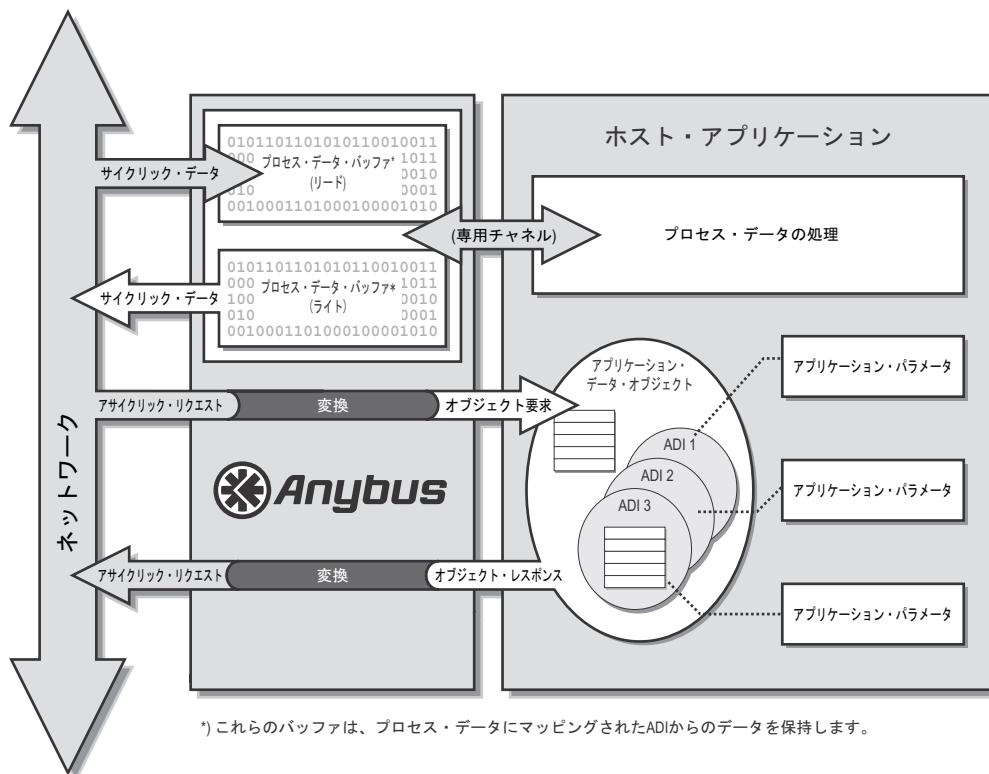
**注意：**上図の'ルータ'部分は、オブジェクト・メッセージ内のヘッダ情報を解釈し、そのデータを適切なソフトウェア機能にルーティングするソフトウェア・コンポーネントを象徴しています。'ドライバ'部分は、ホストの通信を維持することに関して責任があります。この機能は標準の Anybus ファームウェア内で実装されますが、ホスト・アプリケーションでも実装する必要があります。

## 2.3 ネットワーク・データの交換

ネットワーク上で交換されるデータは、アプリケーション・データ・オブジェクト内で分類されます。このオブジェクトはホスト・アプリケーション・ファームウェアで実装する必要があり、その中の各インスタンス（以下 'ADI'（アプリケーション・データ・インスタンス）と呼ぶ）は、ネットワーク・データのブロックを表します。

ADI は通常ではネットワーク側のアサイクリック・パラメータに関連付けられます。例えば、DeviceNet では ADI は専用の CIP オブジェクトを通して表されますが、PROFIBUS ではアサイクリック DP-V1 リード/ライト・サービスを通して ADI にアクセスします。

また、ADI は、ホスト・アプリケーションまたはネットワーク（該当する場合）から、プロセス・データとしてマッピングできます。プロセス・データは Anybus-CompactCom ホスト・プロトコル内の専用データ・チャンネルを通して交換されますが、通常は高速のサイクリック・ネットワーク I/O に関連付けられています。プロセス・データの正確な表示は、ネットワークに特有のものです。例えば、PROFIBUS では、プロセス・データは IO データと相互に関連しています。



各 ADI には名前、データ・タイプ、範囲、およびデフォルト値がタグ付けされており、これらは全てネットワークからアクセスできます（問題になっているネットワークによってサポートされている場合）。これによって、より高いレベルのネットワーク・デバイス（ネットワーク・マスター、コンフィグレーション・ツールなど）は、その実際の名前とタイプ（該当する場合）によってアサイクリック・パラメータを認識することができ、ネットワーク設定プロセスを簡略化できます。

一部のネットワーキング・システムでは、同じパラメータに対するサイクリックおよびアサイクリック・アクセスが可能です。Anybus-CompactCom の場合、これは、ADI が明示的オブジェクト要求とプロセス・データを介して同時にアクセスできることを意味します。Anybus モジュールはこのデータの同期化を試みません。必要な場合は、このシナリオを処理するために必要なメカニズムを、ホスト・アプリケーションが実装する必要があります。

Anybus インターフェースは、リトル・エンディアンのメモリ・アドレス指定を使用します。これは、最下位のバイト (LSB) から最上位のバイト (MSB) へのバイト・オーダーであることを意味します。ただし、Anybus は、実際のネットワーク表示 (初期化中にアプリケーションに対して示される) に応じて、ADI の値を透過的に処理します。アプリケーション・ドライバは、バイト・スワップに関与しています (必要な場合)。このアプローチの使用は、次の理由により決定されます。

- Anybus が、メモリの制限と起動時間の要求によって、全ての ADI のデータ・タイプに関する情報を保持できない
- 全てのパラメータのリードまたはライト要求の前にデータ・タイプをリードする代替方法に時間がかかりすぎる

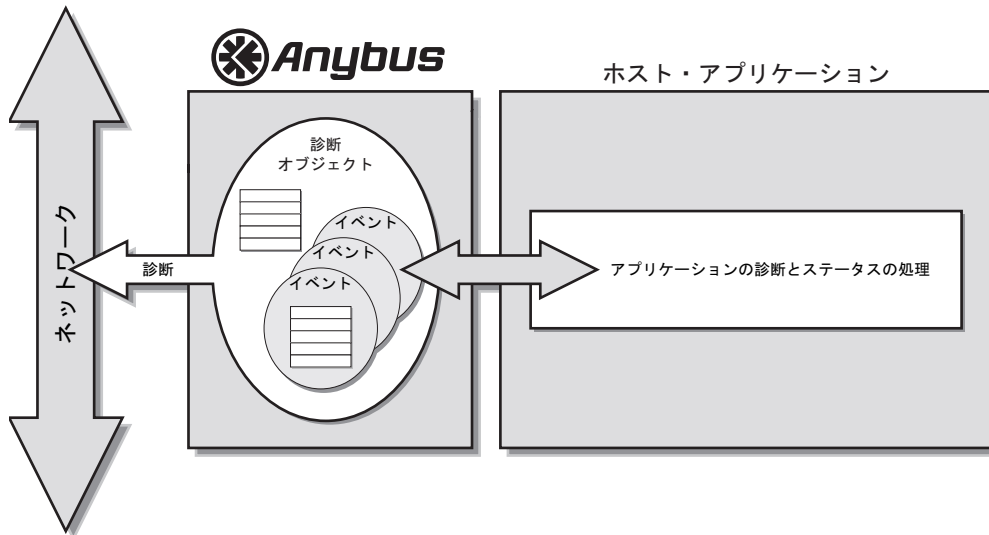
下記も参照してください。

- 14 ページの “プロセス・データ・サブフィールド”
- 23 ページの “Anybus ステート・マシン”
- 46 ページの “ネットワーク・オブジェクト (03h)”
- 54 ページの “アプリケーション・データ・オブジェクト (FEh)”

## 2.4 診断

Anybus-CompactCom は、ホスト関連の診断の専用オブジェクトを特徴としています。診断イベントを報告するには、ホスト・アプリケーションはこのオブジェクト内にインスタンスを作成する必要があります。イベントが解決すると、ホストは単純に診断インスタンスを再び削除します。

各イベントには、イベントの性質を指定するイベント・コードとイベントの重大度を指定する重大度コードがタグ付けされます。この情報の実際の表示は、ネットワークに特有のものであります。



下記も参照してください。

- 43 ページの “診断オブジェクト (02h)”

## 2.5 多言語によるサポート

該当する場合、Anybus-CompactCom は複数の言語をサポートします。これは主にインスタンス名と列挙文字列に影響を与えます。また、Anybus オブジェクトの現在の言語設定に基づきます。これはホスト・アプリケーション・オブジェクトにも適用されますので、注意してください。ホスト・アプリケーションで、それに応じて列挙文字列などの言語を変更できる必要があります。

該当する場合、Anybus モジュールは、ネットワークからアプリケーション・オブジェクトに言語変更要求を転送します。要求を許可するか拒否するかはホスト・アプリケーションによって決定され、モジュールはその言語設定を変更するか、または元のネットワーク要求を拒否します。

サポートしている言語：

- 英語 (デフォルト)
- ドイツ語
- スペイン語
- イタリア語
- フランス語

下記も参照してください。

- 63 ページの “アプリケーション・オブジェクト (FFh)”



## 3. ホスト通信レイヤ

### 3.1 概要

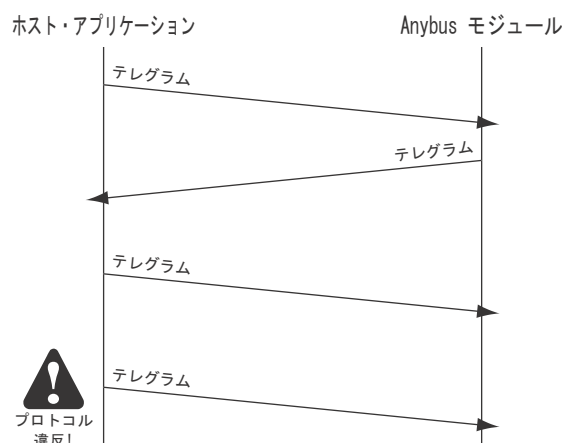
#### 3.1.1 基本原理

その最も低いレベルでは、ホスト・インターフェースの通信は半二重プロトコルに基づいています。テレグラムはピンポン方式で交換され、唯一の例外は、起動中にホスト・アプリケーションによって発行される最初のテレグラムと、送信エラーによって起こりうる再送信（シリアル・ホスト・インターフェースを使用している場合のみ該当）です。

ホスト・アプリケーションと Anybus モジュールは、許容できるスループットを提供するテレグラム交換レートを維持するために同等の義務を共有しますが、ローカル CPU のパフォーマンス能力を超えることはありません。

Anybus モジュールは現在の状態に基づいてできるだけすばやく応答しますが、アプリケーションは独自のタイミング要求を満たすため、または実際のネットワークの要求を満たすために、必要な速さで応答します（いずれか要求の高いほう）。

パラレルおよびシリアル・インターフェースは、各インターフェースのニーズを満たすわずかな相違点（シリアル送信の CRC など）を除き、同じ基本プロトコルを共有します。



#### 3.1.2 テレグラムの内容

各テレグラムは次の3つのサブフィールドで構成されます。

- **ハンドシェーク・レジスタ**

このフィールドは、ホストと Anybus モジュールの通信を制御します。

下記も参照してください。

- 12 ページの“ハンドシェーク・レジスタ”

- **メッセージ・サブフィールド**

このフィールドは、オブジェクト・メッセージング（ホスト・アプリケーションとモジュールの間でメッセージを伝送する組み込みサブプロトコル）に使用されます。このフィールドは全てのテレグラムに存在しますが、その内容は、ハンドシェーク・レジスタ内の特定のビットによって、適切な場合と適切でない場合があります。

下記も参照してください。

- 25 ページの“オブジェクト・メッセージング”

- **プロセス・データ・サブフィールド**

このフィールドは、プロセス・データとして知られる高速のサイクリック・ネットワーク I/O に専用チャネルを提供します。

下記も参照してください。

- 14 ページの“プロセス・データ・サブフィールド”

## 3.2 ハンドシェーク・レジスタ

### 3.2.1 制御レジスタ（リード/ライト）

このレジスタは、Anybus モジュールへの通信を制御します。

b7 (MSB)	b6	b5	b4	b3	b2	b1	b0 (LSB)
CTRL_T	CTRL_M	CTRL_R	CTRL_AUX	-	-	-	-
ビット	説明						
CTRL_T	ホスト・アプリケーションは、新しいテレグラムを送信する場合にこのビットをトグルする必要があります。CTRL_T は、アプリケーションによってモジュールに送信される最初のテレグラムでは "1" に設定する必要があります。						
CTRL_M	設定されている場合、現在のテレグラムのメッセージ・サブフィールドは有効です (18 ページの "メッセージのフラグメンテーション" も参照)。						
CTRL_R	設定されている場合、ホスト・アプリケーションは新しいコマンドを受信する準備ができています (28 ページの "フロー制御" も参照)。						
CTRL_AUX	13 ページの "予備ビット (STAT_AUX、CTRL_AUX)" を参照						
-	(予約、ゼロに設定)						

### 3.2.2 ステータス・レジスタ（リード・オンリー）

このレジスタは、Anybus モジュールの現在のステータスを保持します。

b7 (MSB)	b6	b5	b4	b3	b2	b1	b0 (LSB)
STAT_T	STAT_M	STAT_R	STAT_AUX	SUP	S2	S1	S0
ビット	説明						
STAT_T	モジュールが新しいテレグラムを発行すると、このビットは、ホスト・アプリケーションから最後に受信したテレグラムで、CTRL_T と同じ値に設定されます。						
STAT_M	設定されている場合、現在のテレグラムのメッセージ・サブフィールドは有効です (18 ページの "メッセージのフラグメンテーション" も参照)。						
STAT_R	設定されている場合、Anybus モジュールは着信コマンドを処理する準備ができています (28 ページの "フロー制御" も参照)。						
STAT_AUX	13 ページの "予備ビット (STAT_AUX、CTRL_AUX)" を参照						
SUP	値：意味： 0：モジュールは監視されません。 1：モジュールは別のネットワーク・デバイスによって監視されます。						
S <sub>[0...2]</sub>	これらのビットはモジュールの現在の状態を示します (23 ページの "Anybus ステート・マシン" も参照)。						
	S2	S1	S0	Anybus の状態			
	0	0	0	SETUP			
	0	0	1	NW_INIT			
	0	1	0	WAIT_PROCESS			
	0	1	1	IDLE			
	1	0	0	PROCESS_ACTIVE			
	1	0	1	ERROR			
	1	1	0	(予約)			
	1	1	1	EXCEPTION			

**注意：**ステータス・レジスタは、起動時にクリアされたとみなす必要があります。従って、STAT\_R は有効にクリア (0) されるため、ホスト・アプリケーションによって発行される最初のテレグラムには、有効なメッセージ・サブフィールドを含めてはなりません。

### 3.2.3 監視ビット (SUP)

Anybus ステータス・マシンはサイクリック・データ交換の状態を反映しますが、SUP ビットはアサイクリック通信を含むネットワーク通信の全体的な状態を示します。例えば、CIP では、このビットはマスターがモジュールと接続していることを示します。この接続は、I/O 接続またはアサイクリック (明示的) 接続です。後者の場合、通信は長期間にわたり "無音" で、ステータス・マシンはネットワークがアイドル状態であることを示します。しかし、SUP ビットは、まだモジュールとの接続状態があることを示します。

この機能を処理する方法、提供されるセキュリティのレベル、およびそれを正しく作動させる方法はネットワーク特有の問題であり、しばしば外部の環境 (ネットワークおよび他のネットワーク・デバイスのコンフィグレーションなど) によって異なります。従って、SUP ビットの使用が適切かどうかは、特定の各アプリケーションおよびネットワークによって決定する必要があります。

下記も参照してください。

- 12 ページの “ステータス・レジスタ (リード・オンリー)” (ビット 3、'SUP')

---

**重要:** このビットが各テレグラム送信時のネットワーク状態を反映するということを認識することが重要です。ネットワークからのメッセージ (パラメータ・データ要求や他のコマンドなど) が何らかの理由で遅れた場合 (例えば、アプリケーションがしばらく CTRL\_R ビットをクリアした場合)、アプリケーションは、状況に応じて代替の措置をとらなければならない場合があります。

### 3.2.4 予備ビット (STAT\_AUX、CTRL\_AUX)

デフォルトではこのビットは使用されていないため、ゼロでなければなりません。オプションとして、Anybus オブジェクトでこのビットの追加機能を指定できます (インスタンス・アトリビュート #15、'Auxiliary Bit')。

ライトした時点で、次の機能が定義されました。

- **デフォルト**  
使用しません。STAT\_AUX および CTRL\_AUX はゼロとします。
- **変更データ表示**  
このモードでは、CTRL\_AUX および STAT\_AUX ビットは、現在のテレグラムのプロセス・データが以前のデータと比較して変化したかどうかを示します。

下記も参照してください。

- 12 ページの “制御レジスタ (リード / ライト)” (ビット 4、'CTRL\_AUX')
- 12 ページの “ステータス・レジスタ (リード・オンリー)” (ビット 4、'STAT\_AUX')
- 14 ページの “プロセス・データ・サブフィールド” (3-15 “変更データ表示”)
- 39 ページの “Anybus オブジェクト (01h)” (インスタンス・アトリビュート #15)

## 3.3 プロセス・データ・サブフィールド

### 3.3.1 概要

プロセス・データには2つの種類があります。リード・プロセス・データとライト・プロセス・データです。

- **リード・プロセス・データ**  
リード・プロセス・データはネットワークから受信したデータを表します。Anybus モジュールから受信したテレグラムに存在します（使用されている場合）。
- **ライト・プロセス・データ**  
ライト・プロセス・データはネットワークに送信するデータを表します。Anybus モジュールに送信したテレグラムに存在します（使用されている場合）。

プロセス・データがネットワーク上でどのように表されるかは、例えば PROFIBUS では IO データとして交換され、CANopen では PDO を通して交換されるなど、場合によって異なります。

下記も参照してください。

- 23 ページの“Anybus ステート・マシン”

#### 重要：

- プロセス・データ・サブフィールドは、モジュールが完全に初期化されるまで存在しません。これは一般的にパラレル・ホスト・インターフェースを使用する場合には問題になりませんが、シリアル・ホスト・インターフェースを使用する場合には、テレグラムに影響を与えるということに注意する必要があります。
- Anybus モジュールは、このデータではバイト・スワップを行いません。従って、ホスト・アプリケーションは、実装によって必要とされる場合にのみバイト・スワップに関与します。

### 3.3.2 プロセス・データのマッピング

プロセス・データ・サブフィールドの実際のサイズと構造はプロセス・データのマップに基づいており、起動中およびネットワークからの特定の Anybus の実装の場合に、ホスト・アプリケーションによって指定できます。

プロセス・データのマッピングは、基本的により高いレベルでサイクリック I/O のサイズを定義する構造化された方法で、ホスト・アプリケーションがバイナリ・データの匿名ブロックを指定する代わりに、その実際のデータ・タイプによってデータ構造などを定義できるようにします。

下記も参照してください。

- 8 ページの“ネットワーク・データの交換”
- 46 ページの“ネットワーク・オブジェクト (03h) ”
- 48 ページの“コマンドの詳細：Map\_ADI\_Write\_Area”
- 49 ページの“コマンドの詳細：Map\_ADI\_Read\_Area”
- 54 ページの“アプリケーション・データ・オブジェクト (FEh) ”
- 60 ページの“コマンドの詳細：Remap\_ADI\_Write\_Area”
- 62 ページの“コマンドの詳細：Remap\_ADI\_Read\_Area”

### 3.3.3 変更データ表示

**注意：**この機能は任意で選択できます。デフォルトでは有効になっていません。

任意で、予備ビット (STAT\_AUX と CTRL\_AUX) を使用することにより各テレグラムで変更されたデータを示すことができます。この機能が有効になると、これらのビットは、それぞれリード / ライト・プロセス・データの変更データ信号として個別に使用されません。以下に説明します。

- **リード・プロセス・データの変更データ表示**

Anybus モジュールは、リード・プロセス・データの変更について、次のように示します。

- STAT\_AUX が、リード・プロセス・データを含む最後のテレグラムとは異なるリード・プロセス・データを含むテレグラムで (1) に設定されている。
- STAT\_AUX が、リード・プロセス・データを含む最後のテレグラムと等しいリード・プロセス・データを含むテレグラムでクリア (0) されている。
- リード・プロセス・データを含む最初のテレグラムが、全てのゼロを含む '架空の' 以前のプロセス・データ・バッファと比較される。
- STAT\_AUX が、Remap\_ADI\_Read\_Area によって修正されたリード・プロセス・データのマップを含む最初のテレグラムで常に設定される。
- ホスト・アプリケーションが、STAT\_AUX がクリア (0) されたテレグラムのリード・プロセス・データを廃棄することを選択する場合があります。

- **ライト・プロセス・データの変更データ表示**

ホスト・アプリケーションは、ライト・プロセス・データの変更について、次のように示します。

- CTRL\_AUX が、ライト・プロセス・データを含む最後のテレグラムとは異なるライト・プロセス・データを含むテレグラムで (1) に設定されている。
- CTRL\_AUX が、ライト・プロセス・データを含む最後のテレグラムと等しいライト・プロセス・データを含むテレグラムでクリア (0) されている。
- ライト・プロセス・データを含む最初のテレグラムが、全てのゼロを含む '架空の' 以前のプロセス・データ・バッファと比較される。
- CTRL\_AUX が、Remap\_ADI\_Write\_Area によって修正されたライト・プロセス・データのマップを含む最初のテレグラムで常に設定される。
- Anybus モジュールで余分の負荷をかける可能性がある場合でも、ライト・プロセス・データが変更されていないテレグラムで CTRL\_AUX を (1) に設定することが許容される。

下記も参照してください。

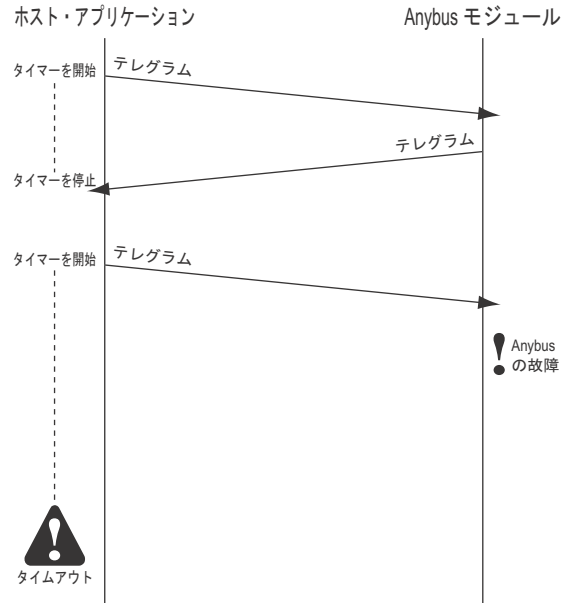
- 13 ページの “予備ビット (STAT\_AUX、CTRL\_AUX) ”
- 39 ページの “Anybus オブジェクト (01h) ” (インスタンス・アトリビュート #15)
- 60 ページの “コマンドの詳細：Remap\_ADI\_Write\_Area”

### 3.4 Anybus ウォッチドッグ

ホスト・アプリケーションは、テレグラムのレスポンス時間を測定することによって、モジュールが適切に機能しているかどうかを確認できます。この時間が特定の値を超えると、モジュールは誤動作していると想定できます。ホスト・アプリケーションは安全な状態になるか、モジュールをリセットするか、または類似した動作をとります。推奨最小タイムアウト値：

インターフェース	最小タイムアウト値
パラレル	10ms
シリアル、19.2kbps	1050ms
シリアル、57.6kbps	360ms
シリアル、115.2kbps	180ms
シリアル、625kbps	60ms

**注意：**これらの値は、全ての場合で Anybus モジュールによって満たされると保証された、推奨最小値にすぎません。特定のアプリケーションに適した値は、安全性要求、シリアル通信の品質（該当する場合）などによって異なります。

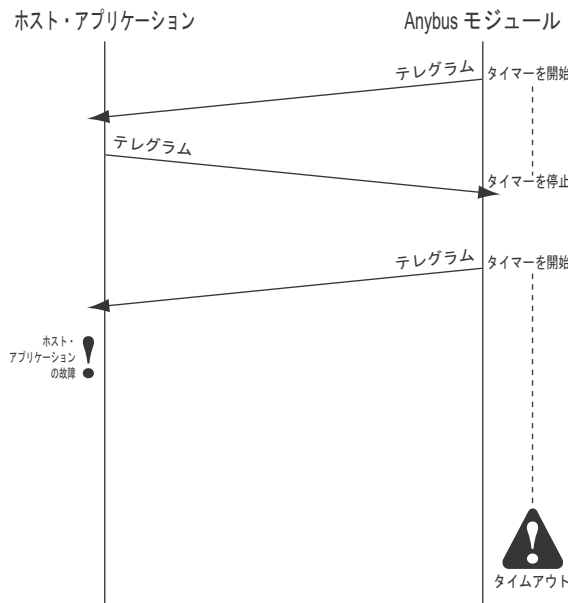


### 3.5 アプリケーション・ウォッチドッグ

有効な場合、Anybus モジュールは、この機能を使用してレスポンス時間を測定することにより、ホスト・アプリケーションが適切に機能しているかどうかを確認できます。この時間が指定された値を超えた場合、Anybus モジュールはホスト・アプリケーションが機能していないとみなし、それに応じた措置を取ります。

タイムアウトが発生した場合に実行される動作はネットワーク特有ですが、モジュールの 'EXCEPTION' 状態への移行は、全ての Anybus-CompactCom の実装に共通しています。

**注意：**シリアル・ホスト・インターフェースを使用する場合、ウォッチドッグ・タイマーはモジュールが新しい送信を開始したときに開始し、モジュールが再送信でない有効なテレグラムを受信したときに停止します。



下記も参照してください。

- 39 ページの “Anybus オブジェクト (01h) ” (インスタンス・アトリビュート #4、'Application watchdog timeout')

## 3.6 シリアル・ホスト・テレグラム

### 3.6.1 概要

シリアル・ホスト・インターフェースでは、テレグラムは共通のアサイクリック・パラレル・インターフェースを通して送信されます。ボーレートは、モジュールのホスト・インターフェース・コネクタでの特定の信号によって決定されます。詳細については、"Anybus CompactCom Hardware Design Guide" を参照してください。

他の通信設定は次の値に固定されています。

データ・ビット： 8  
 パリティ： なし  
 ストップ・ビット：1

**重要：**ほとんどの通信デバイスと同様に、Anybus-CompactCom で使用される実際のボーレートは、理想的なボーレートとは多少異なる場合があります。Anybus モジュールのボーレートのずれは±1.5% 未満です。正常な動作を保証するには、ホスト・アプリケーションでのボーレートのずれは±1.5% を超えてはなりません。

### 3.6.2 シリアル・テレグラム・フレーム

1 バイト	16 バイト	最大 256 バイト	2 バイト
ハンド シェーク・ レジスタ・ フィールド	メッセージ・ サブフィールド	プロセス・データ・ サブフィールド	CRC16
(最初の バイト)			(最後の バイト)

- ハンドシェーク・レジスタ・フィールド**  
 このフィールドには、Anybus モジュールに送信されたテレグラム内の制御レジスタと Anybus モジュールから受信したテレグラム内のステータス・レジスタが含まれます。  
 下記も参照してください。
  - 12 ページの“ハンドシェーク・レジスタ”
- メッセージ・サブフィールド**  
 プロセス・データのスループットを維持するために、メッセージ・サブフィールドは、シリアル・インターフェースを使用する場合、16 バイトに制限されます。それよりも長いメッセージは、小さいいくつかのフラグメントとして交換されます (18 ページの“メッセージのフラグメンテーション”を参照)。  
 下記も参照してください。
  - 25 ページの“オブジェクト・メッセージング”
- プロセス・データ・サブフィールド<sup>1</sup>**  
 このフィールドには、Anybus モジュールに送信されるテレグラムのライト・プロセス・データと、Anybus モジュールから受信するリード・プロセス・データが含まれます。  
 下記も参照してください。
  - 14 ページの“プロセス・データ・サブフィールド”
- CRC16**  
 このフィールドは、16 ビット・サイクリック冗長検査を保持します (Motorola フォーマット、MSB が先)。CRC は、CRC 自体を除いたテレグラム全体を対象としています。  
 下記も参照してください。
  - 82 ページの“CRC の計算”

1. Anybus モジュールが 'SETUP' 状態で動作している場合、このフィールドは存在しません。

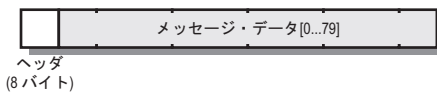
### 3.6.3 メッセージのフラグメンテーション

プロセス・データのスループットを常に保証するために、16 バイトより長いメッセージは、複数の小さい (16 バイト以下の) フラグメントとして送信されます。これらのフラグメントは受信側によって組み立てられ、完全なメッセージとして処理されます。受信者は、メッセージを処理する前に、メッセージ全体が受信されるまで待つ必要があります。これには、最初のフラグメントで検出された可能性がある基本的なエラーが含まれます (データ・フィールドが大きすぎる、エラー・ビットが設定されたコマンド (E=1) である、など)。

各メッセージの後ろには、メッセージが完了したことを受信者に示すために、" 空の " 追加フラグメント (CTRL\_M または STAT\_M=0 のテレグラム) を続ける必要があります。

**例**

メッセージ・フレーム :



80 バイトのメッセージ・データを持つメッセージは、6つの小さいフラグメントとしてモジュールに送信されます。

"空の" 追加フラグメントがメッセージを終了します。

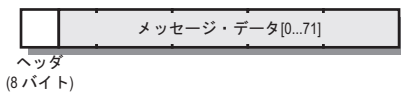
フラグメンテーションのシーケンス :



断片化されたメッセージは、" 空の " フラグメント (CTRL\_M または STAT\_M=0 のテレグラム) を発行して受信側でフラグメンテーション・エラーを発生させることにより、いつでも中断できます。

**例**

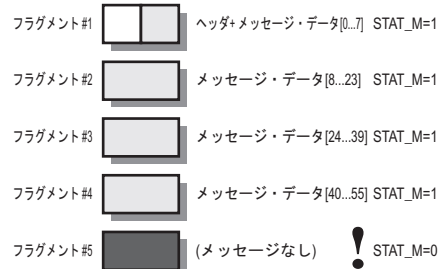
メッセージ・フレーム :



72 バイトのメッセージ・データを持つメッセージは、いくつかの小さいフラグメントとしてモジュールによって送信されます。

メッセージは、" 空の " フラグメントを使用して時期尚早に中断されます。

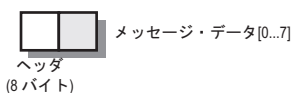
フラグメンテーションのシーケンス :



小さいメッセージ (16 バイト以下) はフラグメンテーションなしで送信されますが、メッセージ全体が完了したことを受信者に通知するために、" 空の " 追加フラグメントを送信する必要があります。

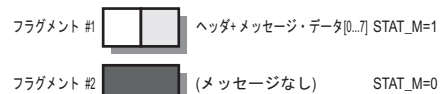
**例**

メッセージ・フレーム :



小さいメッセージ (16 バイト) がホスト・アプリケーションに送信されます。" 空の " 追加フラグメントがメッセージを終了します。

フラグメンテーションのシーケンス :





### 3.6.4 送信エラー

送信エラーは次のように処理されます。

- 受信エラー（CRC の不一致）の場合、テレグラムは無視されます。
- 送信エラーが発生すると、タイムアウト後、最後のテレグラムがホスト・アプリケーションによって再送信されます（異なるボーレートに対して推奨される再送信タイムアウト値を下の表に示す）。
- モジュールは CTRL\_T を監視することによって再送信を検出します。

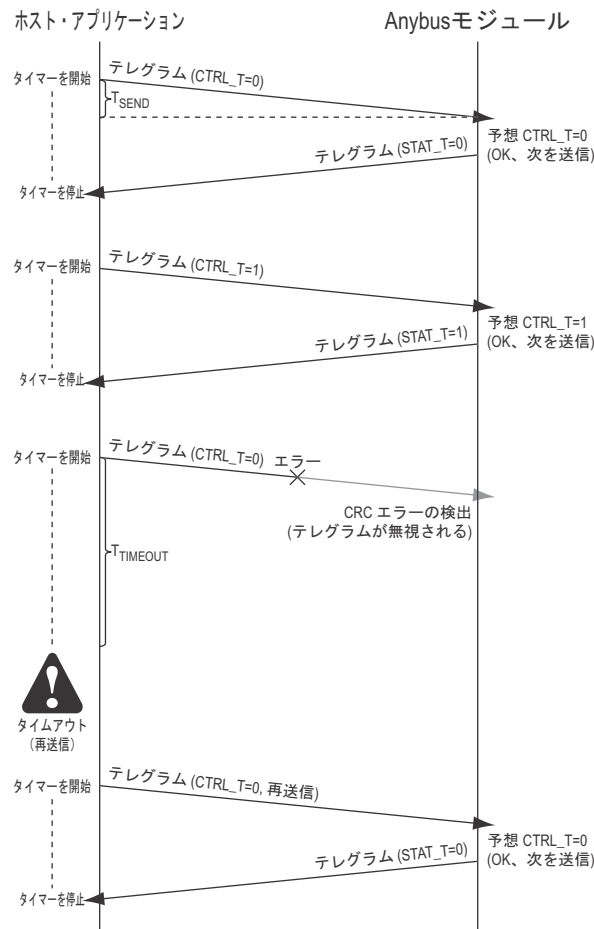
このコンセプトを成功させるには、ホスト・アプリケーションは次の要件を満たす必要があります。

- テレグラムの送信時間は、 $T_{SEND}$  より短くなければなりません（表を参照）。
- 再送信のタイムアウトは、 $T_{TIMEOUT}$  より短くなってはなりません（表を参照）。

ボーレート	$T_{SEND}$	$T_{TIMEOUT}$
19.2kbps	175ms	350ms
57.6kbps	60ms	120ms
115.2kbps	30ms	60ms
625kbps	6ms	20ms

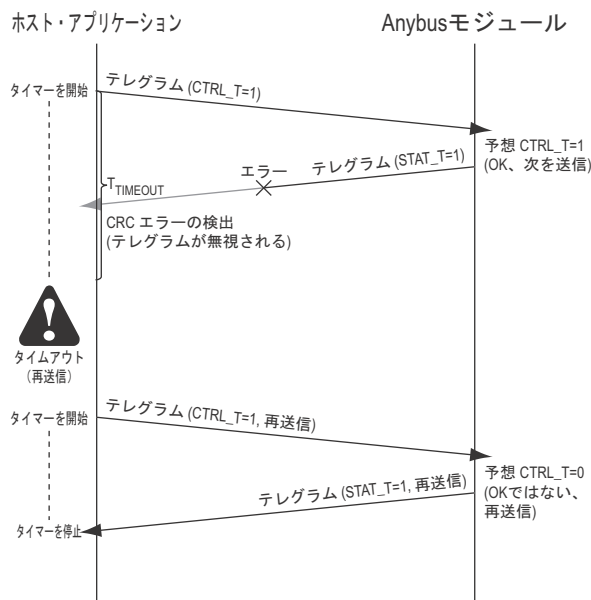
#### シナリオ 1 : Anybus が CRC エラーを検出した場合

この例では、Anybus モジュールが CRC の不一致に遭遇します。エラーが発生したテレグラムは単純に無視され、ホスト・アプリケーションでタイムアウトが発生し、最終的にホスト・アプリケーションに元のテレグラムを再送信させます。



### シナリオ 2 : ホストがエラーを検出した場合

この例では、ホストが CRC の不一致に遭遇します。Anybus モジュールと同様に、ホストはエラーが発生したテレグラムを無視し、最終的にタイムアウトを発生させ、再送信を生成する必要があります。



**重要：**再送信をトリガするのは実際の CRC エラーではなく、タイムアウト自体であることに注意してください。CRC エラーを検出した直後に再送信することはできません。

## 3.7 パラレル・ホスト・テレグラム

### 3.7.1 概要

パラレル・ホスト・インターフェースを使用する場合、テレグラムは共有メモリ領域を経由して交換されます。テレグラム・サブフィールドには、あらかじめ定義されたメモリ位置を経由して簡単にアクセスできます（下記のメモリ・マップを参照）。

テレグラムの送信は制御レジスタを経由してトリガされ、新しいテレグラムの受信はステータス・レジスタで示されます。これは、制御レジスタにアクセスする前に、プロセス・データおよびメッセージのサブフィールドをライトする必要があることを意味します。

テレグラムの受信を待っている間、ステータス・レジスタのポーリング以外でのパラレル・インターフェースへのアクセスは回避する必要があります。

下記も参照してください。

- 12 ページの“ハンドシェイク・レジスタ”

**重要：**内部では、制御レジスタが変更されるたびに、モジュールで割り込みがトリガされます。従って、モジュールはこれを複数のアクセスとして解釈する可能性があるため、このレジスタではビット操作または他のリード/修正/ライト命令を直接使用しないことが重要です。代わりに、全てのビット操作などを一時レジスタで行い、ライトしなおす必要があります。

### 3.7.2 メモリ・マップ

下記に示すアドレス・オフセットは、モジュールのベース・アドレスを基準としています（パラレル・インターフェースが実装されたホスト・アプリケーションのメモリ・スペース内の領域の始まりから）。

アドレス・オフセット：	領域：	アクセス：	注意：
0000h... 37FFh	(予約)	-	(将来の使用のために予約)
3800h... 38FFh	プロセス・データ・ライト領域	ライト・オンリー	14 ページの“プロセス・データ・サブフィールド”を参照
3900h... 39FFh	プロセス・データ・リード領域	リード・オンリー	14 ページの“プロセス・データ・サブフィールド”を参照
3A00h... 3AFFh	(予約)	-	
3B00h... 3C06h	メッセージ・ライト領域	ライト・オンリー	25 ページの“オブジェクト・メッセージング”を参照
3C07h... 3CFFh	(予約)	-	
3D00h... 3E06h	メッセージ・リード領域	リード・オンリー	25 ページの“オブジェクト・メッセージング”を参照
3E07h... 3FFDh	(予約)	-	
3FFEh	制御レジスタ	リード/ライト	12 ページの“制御レジスタ（リード/ライト）”を参照
3FFFh	ステータス・レジスタ	リード・オンリー	12 ページの“ステータス・レジスタ（リード・オンリー）”を参照

**重要：**C プログラマは必ず共有メモリ領域全体を *volatile* として宣言してください。

### 3.7.3 パラレル・テレグラム処理

パラレル・インターフェースでは、送受信はハンドシェーク・レジスタによって管理されます。

下記も参照してください。

- 12 ページの“ハンドシェーク・レジスタ”

#### テレグラムの送信

テレグラムを送信するには、次の手順を行います。

1. 該当する場合、ライト・プロセス・データをプロセス・データ・ライト領域 (3800h...38FFh) にライトします。
2. 該当する場合、メッセージをメッセージ・ライト領域 (3B00h...3C06h) にライトします。
3. 制御レジスタをアップデートし、送信をトリガします。

下記も参照してください。

- 12 ページの“制御レジスタ (リード/ライト)”

#### テレグラムの受信

テレグラムの受信は、ステータス・レジスタの STAT\_T ビットによって合図されます。ホスト・アプリケーションは、ステータス・レジスタをサイクリックにポーリングして新しいテレグラムを検出するか、割り込み操作に依存することができます。

- **割り込み操作 (特に推奨)**  
実装された場合、モジュールが新しいテレグラムを発行するたびに割り込みが生成されます。ステータス・レジスタのサイクリックなポーリングと比較して、この機能はオーバーヘッドを大幅に減少できるため、この機能を使用することは一般的に特に推奨されています。
- **ポーリング操作**  
ホスト・アプリケーションはステータス・レジスタをサイクリックにポーリングし、STAT\_T ビットを検査する必要があります。このビットの値が CTRL\_T ビットの値と等しくなると、モジュールは新しいテレグラムを発行します。

---

**重要：**ステータス・レジスタをポーリングする際に有効な結果を保証するために、2回の連続したリードが一致するまでリードする手順を使用する必要があります。

下記も参照してください。

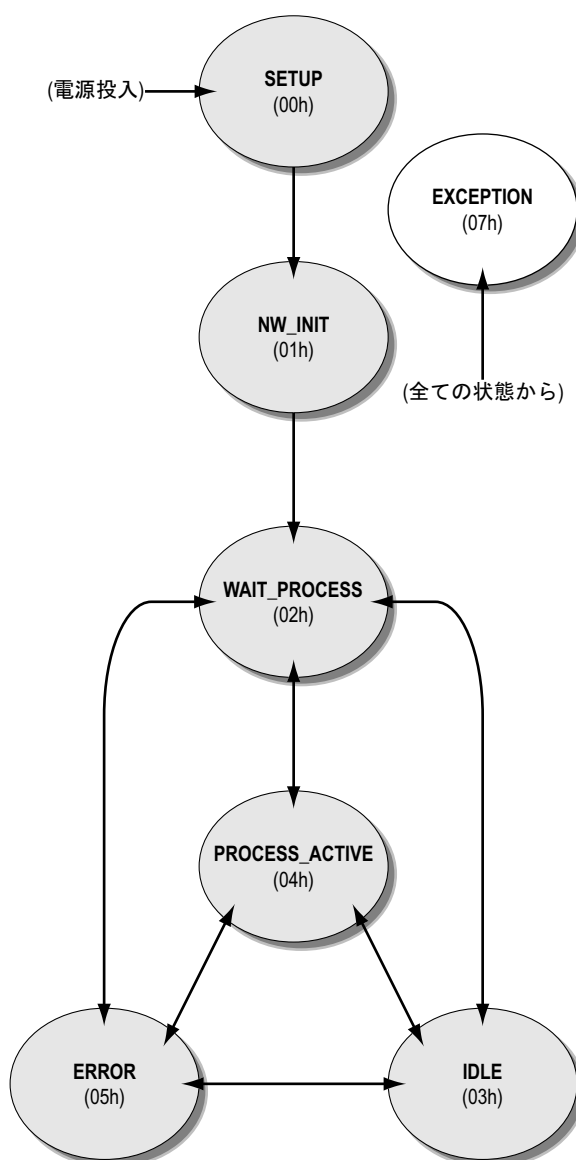
- 12 ページの“ステータス・レジスタ (リード・オンリー)”

## 4. Anybus ステート・マシン

### 4.1 概要

Anybus-CompactCom の基本的な部分は、Anybus ステート・マシンです。特定の時間に、ステート・マシンはモジュールとネットワークのステータスを反映します。

ステート・マシンはムーア・マシンとしてみなされます。ホスト・アプリケーションは全ての状態の移行を記録するには必要ありませんが、それぞれの状態で特定のタスクを行うと予想されています。



下記も参照してください。

- 24 ページの “状態に依存する動作”
- 12 ページの “ステータス・レジスタ (リード・オンリー)”

## 4.2 状態に依存する動作

それぞれの状態に対して予想される動作を次に示します。

状態	説明	予想される動作
SETUP	Anybus のセットアップが進行中です。	36 ページの “Anybus のセットアップ ('SETUP' 状態)” を参照。
NW_INIT	Anybus モジュールは、現在ネットワーク関連の初期化タスクを行っています。 テレグラムには現在プロセス・データが含まれていますが（そのようなデータがマッピングされる場合）、ネットワークのプロセス・データ・チャンネルはまだアクティブではありません。	37 ページの “ネットワークの初期化 ('NW_INIT' 状態)” を参照。
WAIT_PROCESS	ネットワークのプロセス・データ・チャンネルは、一時的に非アクティブです。	ホスト・アプリケーションは、リード・プロセス・データを有効ではないとみなします。
IDLE	ネットワーク・インターフェースはアイドル状態です。この状態の正確な解釈はネットワークにより異なります。ネットワークのタイプによって、リード・プロセス・データは更新されるか静的な状態（変更されない）であるかのいずれかです。	ホスト・アプリケーションはリード・プロセス・データに作用するか、またはアイドル状態になります。
PROCESS_ACTIVE	ネットワークのプロセス・データ・チャンネルはアクティブで、エラーがありません。	正常なデータ処理を行います。
ERROR	少なくとも1つの重大なネットワーク・エラーがあります。	リード・プロセス・データは有効ではないとみなされます。任意で、ホスト・アプリケーションはネットワーク特有の動作を行うことができます。 ライト・プロセス・データはマスターに転送できるので、アプリケーションはこのデータの更新を維持する必要があります。
EXCEPTION	ホスト・アプリケーション関連のエラーのため、モジュールは全てのネットワークへの参加を中止しました。 この状態は回復不能です。ネットワーク・データを交換できるようにするために、モジュールを再起動する必要があります。	可能な場合、エラーを修正します（エラーの詳細情報は Anybus オブジェクトからリードすることができます。39 ページの “Anybus オブジェクト (01h)” を参照）。  完了したら、Anybus モジュールをリセットします。

下記も参照してください。

- 23 ページの “概要”
- 12 ページの “ステータス・レジスタ (リード・オンリー)”

**重要：**ライト・プロセス・データは、Anybus モジュールによってバッファに入れられ、状態移行後にネットワークに送信される場合があるため、ホスト・アプリケーションは、'NW\_INIT' (初期データ)、'WAIT\_PROCESS'、'IDLE'、'ERROR'、および 'PROCESS\_ACTIVE' で、このデータを最新の状態に維持する必要があります。

## 5. オブジェクト・メッセージング

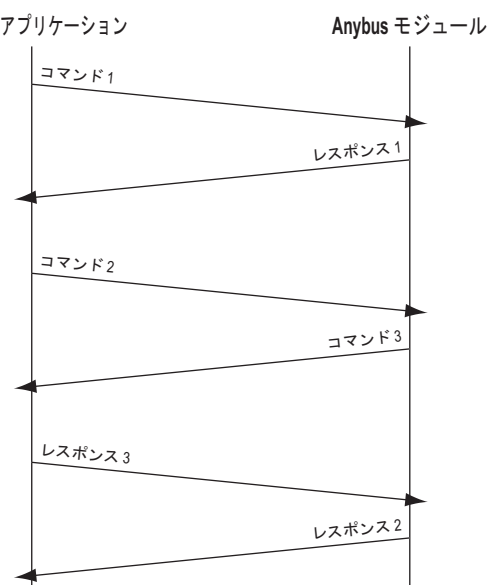
### 5.1 概要

#### 5.1.1 基本原理

オブジェクト・メッセージングには、2つのタイプのメッセージがあります。コマンドとレスポンスです。メッセージ・レベルでは、ホスト・アプリケーションと Anybus モジュールの間には、マスターとスレーブの関係はありません。両方の部分がコマンドを発行でき、また応答する必要があります。コマンドとレスポンスは、常に Anybus オブジェクト・モデル内のインスタンスに関連付けられます。これはオブジェクト自体（インスタンス #0 を通してアドレス指定される）またはその中のインスタンスのいずれかです。

コマンドはいつでも発行できますが（受信側が新しいコマンドを受け入れる準備ができていてという前提で）、レスポンスは以前に受信したコマンドに対する反応としてのみ送信する必要があります。予想外のレスポンスまたは不正なレスポンスは、必ず廃棄してください。

コマンドとレスポンスは非同期で処理されます。新しいコマンドは、以前のコマンドのレスポンスが返送される前に発行される場合があります。また、これはコマンドが到着順で実行されるように保証されておらず、レスポンスが任意の順番で返送されることを意味しています（図を参照）。必要な場合には、ホスト・アプリケーションは、その動作または結果が連続したコマンドに影響を与える可能性があるコマンドのレスポンスを待つ必要があります。



#### 5.1.2 送信元 ID

どのレスポンスがどのコマンドに属するかについての記録を取るために、各メッセージには送信元 ID のタグが付けられます。コマンドを発行する場合、ホスト・アプリケーションは送信元 ID を任意に選択できますが、Anybus モジュールによって発行されるコマンドに応答する場合、レスポンスの送信元 ID は元のコマンドからコピーする必要があります。

下記も参照してください。

- 26 ページの“メッセージのレイアウト”

#### 5.1.3 エラーの処理

何らかの理由でコマンドを処理できない場合でも、受信者はレスポンスを提供する必要があります。このような場合、メッセージ・データ・フィールド内の適切なエラー・コードとともに、レスポンス・メッセージのヘッダ内にエラーのフラグが立てられます。

コマンド・イニシエータはレスポンスを検査し、そのレスポンスがコマンドに対して成功したレスポンスかエラー・メッセージかを確認する必要があります。

下記も参照してください。

- 26 ページの“メッセージのレイアウト”
- 30 ページの“エラー・コード”

## 5.2 メッセージのレイアウト

オブジェクト・メッセージは、後ろにメッセージ関連のデータが続く 8 バイトのヘッダで構成されています。

オフセット	内容							説明
	b7	b6	b5	b4	b3	b2	b1	
0	送信元 ID							25 ページの “送信元 ID” を参照
1	オブジェクト							Anybus オブジェクト・モデル内の送信元 / 宛先を指定します。
2	インスタンス (lsb)							
3	インスタンス (msb)							
4	E	[Command Code]						<b>値：意味：</b> 0：メッセージはコマンドまたは成功したレスポンスのいずれかです。 1：メッセージはエラー・レスポンスです。
	C							<b>値：意味：</b> 0：メッセージはレスポンスです。 1：メッセージはコマンドです。
	コマンド・コード							29 ページの “コマンドの仕様” を参照
5	メッセージ・データ・サイズ							MsgData[] フィールドのサイズ (バイト) (最大 255 バイト)
6	CmdExt[0]							コマンド特有の拡張機能。29 ページの “コマンドの仕様” を参照。
7	CmdExt[1]							
8...n	MsgData[0...n]							メッセージ・データ・フィールド



## 5.3 データ・フォーマット

### 5.3.1 使用できるデータ・タイプ

Anybus-CompactCom は、次のデータ・タイプを標準として使用します。追加のネットワーク特有のデータ・タイプについては、各 Network Interface Appendix で説明しています（該当する場合）。

#	タイプ	ビット	説明	範囲	全てのネットワークで使用可能 <sup>a</sup>	プロセス・データに有効 <sup>b</sup>
0	BOOL	8	ブーリアン	0 = 偽、!0 = 真	はい	はい
1	SINT8	8	符号付きの 8 ビット整数	-128... +127	はい	はい
2	SINT16	16	符号付きの 16 ビット整数	-32768...+32767	はい	はい
3	SINT32	32	符号付きの 32 ビット整数	$-2^{31} \dots +(2^{31}-1)$	はい	はい
4	UINT8	8	符号なしの 8 ビット整数	0...+255	はい	はい
5	UINT16	16	符号なしの 16 ビット整数	0...+65535	はい	はい
6	UINT32	32	符号なしの 32 ビット整数	0...+( $2^{32}-1$ )	はい	はい
7	CHAR	8	文字 (ISO 8859-1) <sup>ode</sup>	0...+255	はい	いいえ
8	ENUM	8	列挙 <sup>f</sup>	0...+255	はい	はい
16	SINT64	64	符号付きの 64 ビット整数	$-2^{63} \dots +(2^{63}-1)$	いいえ	はい
17	UINT64	64	符号なしの 64 ビット整数	0...+( $2^{64}-1$ )	いいえ	はい
18	FLOAT	32	浮動小数点 (IEC 60559)	$\pm 1.17549435E-38$ ... $\pm 3.40282347E+38$	いいえ	はい

- この欄では、そのデータ・タイプが全てのネットワークで表示できるかどうかを示します。
- この欄では、そのデータ・タイプがプロセス・データに使用できるかどうかを示します。
- タイプ CHAR の配列は、ネットワークのネイティブ文字列タイプに変換されます。
- 'Set\_Indexed\_Attribute' と 'Get\_Indexed\_Attribute' は、このデータ・タイプには使用できません。
- このデータ・タイプはプロセス・データには使用できません。
- タイプ ENUM のデータは列挙型で、ゼロで始まる連続した範囲の値に限定されます。

### 5.3.2 Char の配列の処理（文字列）

**注意：**このセクションは、主に ADI で CHAR の配列を使用する場合に適用できます。タイプ CHAR の配列は、ネイティブ文字列タイプに変換されます（該当する場合）。最大文字列長さおよびそれを保存するために必要なバッファ・スペースは、データ・タイプとエレメントの数によって定義されます。

CHAR の配列の全てのエレメントは重要です。Anybus モジュールはリードする際に終端文字を予測せず、ライトする際に終端文字を生成しません。文字列の実際の長さは、'Get\_Attribute' および 'Set\_Attribute' コマンドで与えられるペイロードのサイズで定義されます。

一般的に、'エレメントの数'、'データ・タイプ'、およびメッセージ・ペイロードの長さをできるだけ一貫した状態に維持することが推奨されます。Anybus モジュールがペイロードの長さを実際のバッファ・スペースの間の一貫性を確認する保証はありません。

下記も参照してください。

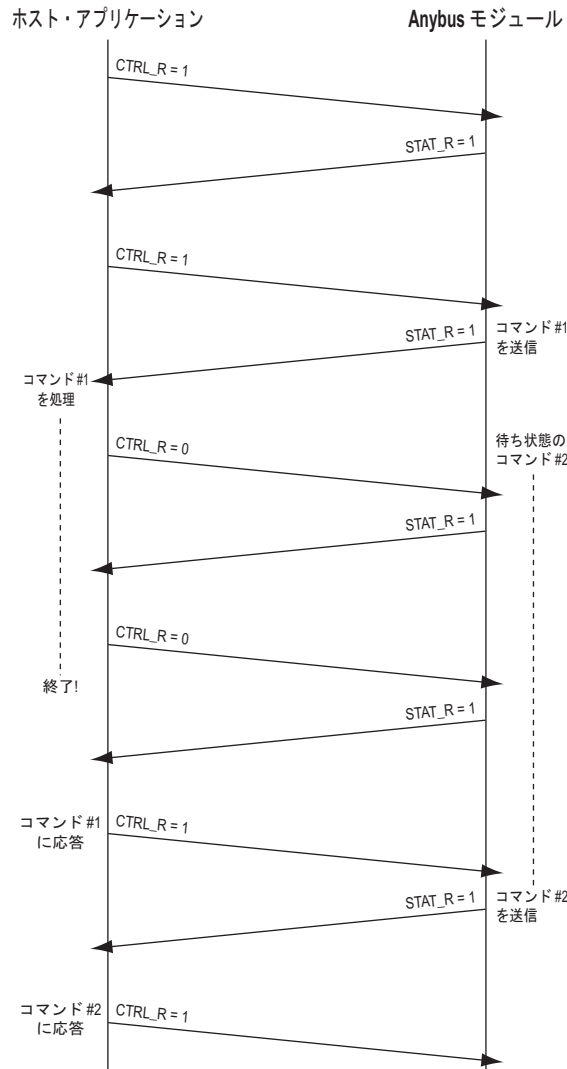
- 54 ページの“アプリケーション・データ・オブジェクト (FEh)”

## 5.4 フロー制御

ホスト・アプリケーションまたは Anybus モジュールがビジー状態であるか、その他の何らかの理由で、新しいコマンドを処理できない場合があります。このような状況に対処するために、CTRL\_R と STAT\_R は、新しいコマンドを処理する準備ができているかどうかを合図する可能性があることを受信側に伝えるために使用されます。

次の規則が適用されます。

- レスポンスはこれらのビットによってブロックされません。コマンドを発行する場合、ホスト・アプリケーションは、常にそのコマンドに対するレスポンスを処理する十分な空きリソースを持っている必要があります。
- CTRL\_R と STAT\_R はメッセージ間でのみチェックされます。シリアル・ホスト・インターフェースを使用する場合、これらのビットを使用して、進行中のメッセージのフラグメンテーションを一時停止または中止することはできません。
- STAT\_R がクリア (0) されている間にホスト・アプリケーションがコマンドを発行した場合、そのコマンドはモジュールによって廃棄され、DC エラー・カウンタは増加します。



下記も参照してください。

- 12 ページの “制御レジスタ (リード / ライト)”
- 12 ページの “ステータス・レジスタ (リード・オンリー)”
- 39 ページの “Anybus オブジェクト (01h)” (インスタンス・アトリビュート #8、'Error counters')

## 5.5 コマンドの仕様

### 5.5.1 概要

この章では、グローバル・コマンド（アクセスしているオブジェクトにかかわらず、同じコマンド・コードを持っているコマンド）について説明します。

一部のオブジェクトには特別の要件があり、オブジェクト特有のコマンドを通して処理されます。このような場合、グローバル・コマンドとは異なり、同じコマンド・コードがコンテキスト（どのオブジェクトにアクセスしているか）によって異なる意味を持つ場合があります。オブジェクト特有のコマンドについては、各オブジェクトに関して個別に説明します（該当する場合）。

下記も参照してください。

- 38 ページの“Anybus モジュール・オブジェクト”
- 52 ページの“ホスト・アプリケーション・オブジェクト”

汎用コマンド記述に関して、コマンドが定義済みの汎用の記述と構造を持つ場合、その実際の影響はコンテキストによって大きく異なる可能性があることに注意してください。

例えば：

- アプリケーションが Reset を発行→ネットワーク・コンフィグレーション・オブジェクト = ネットワーク設定をリセット
- ネットワークのリセット→Anybus が Reset を発行→アプリケーション・オブジェクト = Anybus が EXCEPTION に移行し、ハードウェアのリセットを待つ

---

**重要：** '予約' のマークが付けられたフィールドは、注意して処理する必要があります。Anybus モジュールに送信されたメッセージの予約されたフィールドは、将来の Anybus の改定で用途を定義される可能性があるため、0（ゼロ）に設定する必要があります。Anybus モジュールから受信されたメッセージでは、予約されたフィールドは単純に無視されません。

## 5.5.2 コマンド・コード

次のコマンドはグローバルです。アクセスしているオブジェクトにかかわらず、同じコマンド・コードが使用されます。

コマンド・コード	コマンド名	ページ
00h	(予約)	-
01h	Get_Attribute	31 ページ
02h	Set_Attribute	31 ページ
03h	Create	32 ページ
04h	Delete	32 ページ
05h	Reset	33 ページ
06h	Get_Enum_String	33 ページ
07h	Get_Indexed_Attribute	34 ページ
08h	Set_Indexed_Attribute	34 ページ
09h... 0Fh	(予約)	-
10h... 30h	(オブジェクト特有のコマンド用に予約)	-
31h... 3Eh	(予約)	-
3Eh... 3Fh	(オブジェクト特有のコマンド用に予約)	-

## 5.5.3 エラー・コード

何らかの理由でコマンドを実行できない場合、レスポンスのメッセージ・データ・フィールド (MsgData[ ]) の最初のバイトは、問題に関する詳細情報をコマンド・イニシエータに提供するために使用されます。

オブジェクト特有の追加エラー情報は、メッセージ・データ・セクションでも追加できます。

値	エラー	意味
00h	(予約)	-
01h		
02h	Invalid message format	コマンドとエラー・ビットが設定されています。
03h	Unsupported object	オブジェクトが登録されていません。
04h	Unsupported instance	ターゲット・インスタンスは存在しません。
05h	Unsupported command	ターゲット・オブジェクトは指定されたコマンドをサポートしていません。
06h	Invalid CmdExt[0]	CmdExt[0] の無効な値、または CmdExt[0] と CmdExt[1] の無効な組み合わせ
07h	Invalid CmdExt[1]	CmdExt[1] での無効な設定
08h	Attribute not Set-able	要求されたアトリビュートは Set できません。
09h	Attribute not Get-able	要求されたアトリビュートは Get できません。
0Ah	Too Much Data	メッセージ・データ・フィールド内のデータが多すぎます。
0Bh	Not Enough Data	メッセージ・データ・フィールドに十分なデータがありません。
0Ch	Out of range	指定された値が範囲外です。
0Dh	Invalid state	コマンドは現在の状態でサポートされていません。
0Eh	Out of resources	リソースが制限されているため、ターゲット・オブジェクトはコマンドを実行できません。
0Fh	Segmentation failure	セグメント化プロトコルの無効な処理
10h	Segmentation buffer overflow	受信したデータが多すぎます。
11h... FEh	(予約)	-
FFh	Object Specific Error	オブジェクトは拡張されたエラー情報を返しました。追加の詳細情報は、メッセージ・データ・フィールド (MsgData[0...n]) に含まれている場合または含まれていない場合があります。

## 5.5.4 Get\_Attribute

### 詳細

コマンド・コード : 01h

有効 : (コンテキストによる)

### 説明

このコマンドは、アトリビュートの値を取得します。

- コマンドの詳細

CmdExt[0]	アトリビュートの番号
CmdExt[1]	(予約)

- レスポンスの詳細

MsgData[0...n]	アトリビュート値
----------------	----------

## 5.5.5 Set\_Attribute

### 詳細

コマンド・コード : 02h

有効 : (コンテキストによる)

### 説明

このコマンドは、アトリビュートに値を割り当てます。

- コマンドの詳細

CmdExt[0]	アトリビュートの番号
CmdExt[1]	(予約)
MsgData[0...n]	アトリビュート値

- レスポンスの詳細

(データなし)

## 5.5.6 Create

### 詳細

コマンド・コード：03h

有効： オブジェクト・インスタンス（インスタンス #0）

### 説明

このコマンドは、オブジェクト内に新規インスタンスを作成します。成功した場合、レスポンスのデータ部分には新しく作成されたインスタンスの番号が含まれます。

- コマンドの詳細  
(オブジェクト特有)
- レスポンスの詳細

MsgData[0]	作成されたインスタンスの番号（ロー・バイト）
MsgData[1]	作成されたインスタンスの番号（ハイ・バイト）

## 5.5.7 Delete

### 詳細

コマンド・コード：04h

有効： オブジェクト・インスタンス（インスタンス #0）

### 説明

このコマンドは、以前に作成されたインスタンスを削除します（上記を参照）。成功した場合、指定されたインスタンスによって占有された全てのリソースが解除されます。

- コマンドの詳細

CmdExt[0]	削除するインスタンス番号（ロー・バイト）
CmdExt[1]	削除するインスタンス番号（ハイ・バイト）

- レスポンスの詳細（成功）  
(データなし)
- レスポンスの詳細（エラー）

Invalid CmdExt[0]	指定されたインスタンスは存在しません。
-------------------	---------------------

## 5.5.8 Reset

### 詳細

コマンド・コード：05h

有効： (コンテキストによる)

### 説明

このコマンドは、オブジェクト上でリセット・コマンドを実行します。

- コマンドの詳細

CmdExt[0]	(予約)
CmdExt[1]	00h = 電源オン・リセット (実際の電源オンまたはシミュレーション) 01h = 工場出荷時設定にリセット 02h = 電源オン+工場出荷時設定にリセット

- レスポンスの詳細

(データなし)

## 5.5.9 Get\_Enum\_String

### 詳細

コマンド・コード：06h

有効： (コンテキストによる)

### 説明

このコマンドは、列挙タイプ (ENUM) であるアトリビュートを取得します。返される値は、指定された列挙値に関連付けられたリテラルの文字列です。

- コマンドの詳細

CmdExt[0]	アトリビュートの番号
CmdExt[1]	列挙値

- レスポンスの詳細 (成功)

MsgData[0...n]	列挙文字列
----------------	-------

- レスポンスの詳細 (エラー)

Invalid CmdExt[1]	列挙値が範囲外です。
-------------------	------------

## 5.5.10 Get\_Indexed\_Attribute

### 詳細

コマンド・コード：07h

有効： (コンテキストによる)

### 説明

このコマンドは、複数のエレメント（配列など）で構成されるアトリビュートの1つのエレメントの値を取得します。このコマンドを、タイプ CHAR のアトリビュートにアクセスするために使用することはできません。

- コマンドの詳細

CmdExt[0]	アトリビュートの番号
CmdExt[1]	インデックス（最初のエレメントにはインデックス0がある）

- レスポンスの詳細（成功）

MsgData[0...n]	値
----------------	---

- レスポンスの詳細（エラー）

Invalid CmdExt[1]	インデックスが範囲外です。
-------------------	---------------

## 5.5.11 Set\_Indexed\_Attribute

### 詳細

コマンド・コード：08h

有効： (コンテキストによる)

### 説明

このコマンドは、複数のエレメント（配列など）で構成されるアトリビュートの1つのエレメントに値を割り当てます。このコマンドを、タイプ CHAR のアトリビュートにアクセスするために使用することはできません。

- コマンドの詳細

CmdExt[0]	アトリビュートの番号
CmdExt[1]	インデックス（最初のエレメントにはインデックス0がある）
MsgData[0...n]	設定する値

- レスポンスの詳細（成功）

(データなし)

- レスポンスの詳細（エラー）

Invalid CmdExt[1]	インデックスが範囲外です。
-------------------	---------------



## 6. 初期化と起動

### 6.1 概要

ネットワークに参加する前に、次のステップを完了する必要があります。

#### 1. 最初のハンドシェーク

最初のハンドシェークの目的は、両方の部分（ホスト・アプリケーションと Anybus モジュール）で通信の準備ができているかを確認することです。準備が完了すると、モジュールは 'SETUP' 状態になります。詳細については、35 ページの“最初のハンドシェーク”を参照してください。

#### 2. Anybus のセットアップ

このステップでは、モジュールの操作方法を決定します。これが完了すると、モジュールは 'NW\_INIT' 状態になります。

詳細については、36 ページの“Anybus のセットアップ ('SETUP' 状態)”を参照してください。

#### 3. ネットワークの初期化

この段階では、モジュールはホスト・アプリケーションからの情報をリードして評価しようとしています。これが完了すると、モジュールは 'WAIT\_PROCESS' 状態になります。

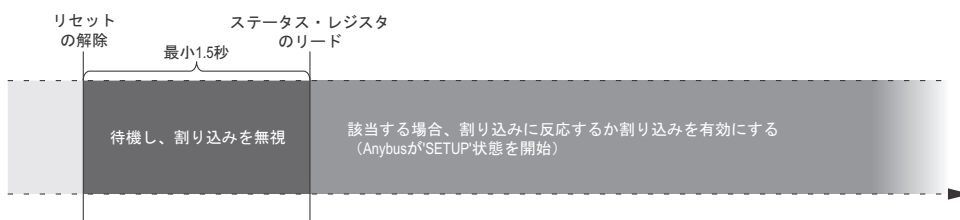
詳細については、37 ページの“ネットワークの初期化 ('NW\_INIT' 状態)”を参照してください。

### 6.2 最初のハンドシェーク

手順は使用するホスト・インターフェースのタイプによって多少異なります。

#### • パラレル・ホスト・インターフェース

ホスト・アプリケーションはリセット後少なくとも 1.5 秒待ってから、ステータス・レジスタをリードする必要があります。この間に検出される割り込みは無視する必要があります。



#### • シリアル・ホスト・インターフェース

ホスト・アプリケーションは、その最初のテレグラムを送信する前に、リセット後少なくとも 1.5 秒待つ必要があります。



**注意：**ステータス・レジスタは、起動時にクリアされたとみなされます。これは、STAT\_T が 0（ゼロ）に設定されているとみなされているため、最初のテレグラムにメッセージ・データを含めてはならないということを意味します。

## 6.3 Anybus のセットアップ ('SETUP' 状態)

この段階には 4 つの特徴的なステップがあります。

### 1. Anybus モジュールに関する情報の収集 (任意)

ホスト・アプリケーションは、Anybus オブジェクト (01h) から、モジュールを設定する際に関連する可能性がある他の特性とともに、ネットワーク・タイプを取得する場合があります。この情報は、モジュール・タイプ値などに基づいて、異なる実装を選択するためにも使用される場合があります。

### 2. ネットワークのコンフィグレーション (任意)

この段階では、ホスト・アプリケーションは、その値が物理スイッチから生じるネットワーク・コンフィグレーション・オブジェクト内の全てのインスタンスを更新する必要があります (ノード・アドレス、ボーレートなど)。キーパッドやディスプレイなどの "ソフト" 入力デバイスから行う設定は、この時点では更新してはなりません。

### 3. プロセス・データのマッピング (任意<sup>1,2</sup>)

ホスト・アプリケーションでは、任意で ADI をプロセス・データにマッピングする場合があります。

### 4. セットアップの確定

セットアップ手順は、Anybus オブジェクト (01h) 内の 'Setup Complete' アトリビュートを TRUE に設定することによって確定します。

成功した場合、モジュールは 'NW\_INIT' 状態に移行し (下記)、失敗した場合には 'EXCEPTION' 状態に移行します。後者の場合、Anybus オブジェクト (01h) 内の 'Exception' アトリビュートから詳細情報をリードすることができます。

下記も参照してください。

- 8 ページの “ネットワーク・データの交換”
- 23 ページの “Anybus ステート・マシン”
- 39 ページの “Anybus オブジェクト (01h)”
- 50 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. このステップは任意ですが、一部のネットワーキング・システムや Anybus の実装で必要な場合があります。
  2. 特定の Anybus の実装 (Anybus-CompactCom Drive Profile 製品系列) では、ランタイム中にプロセス・データのマップの変更を試みる場合があります。詳細については、54 ページの “アプリケーション・データ・オブジェクト (FEh)” を参照してください。

## 6.4 ネットワークの初期化 ('NW\_INIT' 状態)

この段階では、Anybus モジュールはホスト・アプリケーションからの情報をリードして評価しようとしています。ホスト・アプリケーションは、ホスト・アプリケーション・オブジェクトへの着信要求に応答するために必要です。要求されたオブジェクトまたはアトリビュートがホスト・アプリケーションで実装されない場合、単純にエラー・メッセージで応答します。この場合、モジュールは要求されたアトリビュートにそれ自身のデフォルト値を使用します。

ホスト・アプリケーションは、物理スイッチから発生しないインスタンスを含めて、ネットワーク・コンフィグレーション・オブジェクト内のインスタンスを自由に更新します。

この状態で重大なエラー（モジュールの進行を妨げるエラー）が発生した場合、モジュールは 'EXCEPTION' 状態に移行します。Anybus オブジェクト (01h) 内の 'Exception' アトリビュートから詳細情報をリードすることができます。

これが完了すると、モジュールは 'WAIT\_PROCESS' 状態になります。

---

**重要：**この点からのテレグラムにはプロセス・データが含まれる可能性があるため（セットアップによって異なる）、この状態への移行は、特にシリアル・ホスト・インターフェースを使用する場合には不可欠です。このデータはモジュールによってバッファに入れられ、次の状態移行でネットワークに送信される可能性があるため、ライト・プロセス・データをこの状態で更新し続けることは重要です。

下記も参照してください。

- 23 ページの “Anybus ステート・マシン”
- 50 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)”

## 7. Anybus モジュール・オブジェクト

### 7.1 概要

この章のオブジェクトは、全ての Anybus-CompactCom の実装での標準として、実装されています。その機能は、オブジェクトを使用する時期と方法を示すためにカテゴリ化されています。

下記も参照してください。

- 27 ページの“データ・フォーマット”
- 30 ページの“エラー・コード”
- 68 ページの“機能のカテゴリ化”

### 7.2 オブジェクト改定

オブジェクト改定アトリビュートの目的は、Anybus モジュール内のオブジェクトの改定がホスト・アプリケーションでのソフトウェアの実装と互換性があるかどうかをホスト・アプリケーションが決定できるようにし、オブジェクトの改定に基づいて異なる実装を選択できるようにすることです。

原則として、オブジェクト改定は、当該変更によってホスト・アプリケーションのソフトウェア実装で互換性の問題を発生させるような状況で、オブジェクトが変更された場合に更新されます。アトリビュートまたはコマンドが追加されたというような小さい変更は、通常では改定変更の原因にはなりません。

---

**重要：**オブジェクト改定アトリビュートの定義は、開発初期に変更されました。これは、初期の Anybus-CompactCom の実装でのオブジェクト改定が、小さい変更に対しても増加されたことを意味します。該当する場合、現在の定義をどのファームウェア改定から使用するかを記載した注記が各 *Network Interface Appendix* に追加されます。

ご質問がある場合は、HMS のテクニカル・サポート・サービスまでご連絡ください。

## 7.3 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、Anybus モジュール自体に関するデータを組み立てます。問題になっているデータは、モジュールが適応される産業用ネットワークを表すのではなく、モジュールに固有のデータについて説明します。このデータはアプリケーションで使用できます。値は産業用ネットワークによって異なる可能性があり、異なる場合には各アペンディックスで説明されています。

このオブジェクト内のほとんどのアトリビュートには、コマンド `Get_Attribute` を使用してデータを取得できるアクセス・タイプ "get" があります。設定する必要がある唯一のアトリビュートは "Setup Complete" (インスタンス #1、アトリビュート #5) で、セットアップを終了したモジュールを通知するためにアプリケーションによって使用されます。コンフィグレーションが受け入れられない場合、モジュールは EXCEPTION 状態に移行し、情報はインスタンス #1、アトリビュート #6 ("Exception Code") からリードすることができます。

### サポートされているコマンド

オブジェクト : `Get_Attribute` (01h)  
 インスタンス : `Get_Attribute` (01h)  
                   `Set_Attribute` (02h)  
                   `Get_Enum_String` (06h)

### オブジェクトのアトリビュート (インスタンス #0)

#	名前	アクセス	データ・タイプ	値
1	Name	Get	CHAR の配列	'Anybus'
2	Revision	Get	UINT8	04h
3	Number of instances	Get	UINT16	0001h
4	Highest instance no.	Get	UINT16	0001h

### インスタンスのアトリビュート (インスタンス #1)

#	名前	アクセス	カテゴリ	タイプ	説明
1	Module type	Get	Appl. spec. <sup>a</sup>	UINT16	値 :   意味 : 0401h : 標準 Anybus-CompactCom 0402h : Anybus-CompactCom ドライブ・プロファイル (その他) : (将来の製品のために予約)
2	Firmware version	Get	-	構成 : UINT8 メジャー UINT8 マイナー UINT8 ビルド	ファームウェア・バージョン。この値は、一般的に、特定の機能が使用できるかどうかを特定するために使用することはできません。この目的のためには、個々のオブジェクトの 'Revision' アトリビュートを使用してください。
3	Serial number	Get	-	UINT32	一意のシリアル・ナンバー

#	名前	アクセス	カテゴリ	タイプ	説明
4	Application watchdog timeout	Get/Set	拡張	UINT16	アプリケーション・ウォッチドッグのコンフィグレーション 値： 意味： 0： 無効（デフォルト） （その他）：タイムアウト値（ms）  下記も参照してください。 - 3-16 “アプリケーション・ウォッチドッグ”
5	Setup complete	Get/Set	基本	BOOL	このアトリビュートは、Anybus セットアップの段階が完了すると TRUE に設定されます。コンフィグレーションが受け入れられた場合、Anybus モジュールは 'NW_INIT' 状態に移行します。受け入れられない場合（コンフィグレーションで重大なエラーが検出された場合）、モジュールは 'EXCEPTION' 状態に移行します。この場合、'Exception Code' アトリビュート（下記）から詳細情報をリードすることができます。  下記も参照してください。 - 36 ページの “Anybus のセットアップ（'SETUP' 状態）”
6	Exception Code	Get	拡張	ENUM	7-42 “例外コード” を参照。
7	FATAL event	Get/Set	Dev. <sup>b</sup>	構成： （HMS 特有）	最新の FATAL イベント（ある場合）は、このインスタンスのログに記録されます。HMS サポートが評価時に使用します。
8	Error counters	Get	Dev. <sup>c</sup>	構成： UINT16 DC UINT16 DR UINT16 SE UINT16 FE	エラー・カウンタ（FFFFh でカウント停止）： DC： 廃棄されたコマンド（R=0 で受信） DR： 廃棄された（予期しない）レスポンス SE： シリアル受信エラー FE： フラグメンテーション・エラー
9	Language	Get/Set	拡張	ENUM	現在の言語： 値： 列挙文字列： 00h： 'English'（デフォルト） 01h： 'Deutsch' 02h： 'Español' 03h： 'Italiano' 04h： 'Français'  下記も参照してください。 - 63 ページの “アプリケーション・オブジェクト（FFh）” - 67 ページの “コマンドの詳細：Change_Language_Request”

#	名前	アクセス	カテゴリ	タイプ	説明
10	Provider ID	Get	-	UINT16	生産中に HMS によってあらかじめプログラミングされ、FLASH に永続的に保存されます（詳細については、HMS までご連絡ください）。 値： 意味： 0001h : HMS Networks FFFFh : (予約) その他：プロバイダ特有
11	Provider specific info	Get/Set	-	UINT16	このアトリビュートに保存された情報はプロバイダ特有で、あらかじめ定義された意味はなく、Anybus モジュールによって評価されず、使用されません。 このアトリビュートにライトされる値は、不揮発性メモリに保存されます。デフォルト値は 0000h です。
12	LED colors	Get	Appl. spec. <sup>a</sup>	構成： UINT8 LED1A UINT8 LED1B UINT8 LED2A UINT8 LED2B	このアトリビュートは、ネットワーク・ステータス LED の色を指定します。 値： 意味： 00h : なし（使用しない） 01h : グリーン 02h : レッド 03h : イエロー 04h : オレンジ 05h : ブルー 06h : ホワイト
13	LED status	Get	Appl. spec. <sup>a</sup>	UINT8	ビット・フィールドは、ネットワーク・ステータス LED の現在の状態を次のように保持します。 ビット：内容： b0 : LED1A ステータス (0=OFF、1=ON) b1 : LED1B ステータス (0=OFF、1=ON) b2 : LED2A ステータス (0=OFF、1=ON) b3 : LED2B ステータス (0=OFF、1=ON) b4... 7 : (予約)
14	(予約)	-	-	-	-
15	Auxiliary Bit	Get/Set	Appl. spec. <sup>a</sup>	UINT8	下記も参照してください。 - 13 ページの“予備ビット (STAT_AUX、CTRL_AUX)” - 42 ページの“予備ビット”

- このアトリビュートのカテゴリは、アプリケーションでの使用方法によって異なります。
- このアトリビュートの内容は、アプリケーション開発中に HMS サポートへの入力としてのみ使用されます。
- このアトリビュートの内容は、アプリケーション開発中にのみ使用されます。

## 例外コード

EXCEPTION 状態にある場合、このアトリビュートは状態に関する追加情報を提供します。

#	列挙文字列	説明
00h	'No exception'	-
01h	'Application timeout'	ホスト・アプリケーションは、指定されたウオッチドッグ・タイムアウト期間内に応答しませんでした。
02h	'Invalid device address'	選択されたデバイス・アドレスは、実際のネットワークには有効ではありません。
03h	'Invalid communication settings'	選択された通信設定は、実際のネットワークには有効ではありません。
04h	'Major unrecoverable app event'	ホスト・アプリケーションは、診断オブジェクトに対する主要な回復不能なイベントを報告しました。
05h	'Wait for reset'	モジュールはホスト・アプリケーションがリセットを実行するのを待っています。
06h	'Invalid process data config'	プロセス・データのコンフィグレーションは無効です。
07h	'Invalid application response'	ホスト・アプリケーションは、コマンドに対して無効なレスポンスを提供しました。
08h	'Non-volatile memory checksum error'	不揮発性メモリに保存されたパラメータの少なくとも1つは、チェックサム・エラーのためにそのデフォルト値に戻されました。
(その他)	(予約)	-

下記も参照してください。

- 23 ページの “Anybus ステート・マシン”

## 予備ビット

このアトリビュートは、CTRL\_AUX および STAT\_AUX ビットの機能を定義します。

#	機能	CTRL_AUX	STAT_AUX
00h	なし (デフォルト)	機能なし、0 (ゼロ) に設定	機能なし、常に 0 (ゼロ)
01h	変更されたデータの指示	0: ライト・プロセス・データは変更されていません。 1: ライト・プロセス・データは更新されました。	0: リード・プロセス・データは変更されていません。 1: リード・プロセス・データは更新されました。
(その他)	(予約)	-	-

下記も参照してください。

- 3-12 “ハンドシェーク・レジスタ”
- 3-13 “予備ビット (STAT\_AUX、CTRL\_AUX)”

## オブジェクト特有のエラー・コード

次のオブジェクト特有のエラー・コードは、'Setup Complete' アトリビュートの設定に対するレスポンスとして、モジュールによって返される場合があります。

#	エラー	説明
01h	Invalid process data configuration	プロセス・データのコンフィグレーションは無効です。
02h	Invalid device address	選択されたデバイス・アドレスは、実際のネットワークには有効ではありません。
03h	Invalid communication settings	選択された通信設定は、実際のネットワークには有効ではありません。



## 7.4 診断オブジェクト (02h)

### カテゴリ

各産業用ネットワークに特有 (アペンディックスを参照)

### オブジェクトの説明

このオブジェクトは、ネットワークに診断イベントを報告する標準的な方法を示します。ネットワーク上での表示方法は異なりますが、全ての実装では共通して、モジュールは主要な回復不能なイベントの場合に 'EXCEPTION' 状態になります。

モジュールが起動して初期化された場合、このモジュールにはインスタンスが存在しません。アプリケーションで診断イベント (溶断ヒューズなど) が発生した場合、そのアプリケーションはイベントの重大度と種類に関する情報を含むインスタンスを作成します。このインスタンス内の情報は、アプリケーションがインスタンスを削除するまで、アプリケーションで使用できます。インスタンス内のイベント・コードはモジュールによって処理され、イベントに関するネットワーク特有の正しい情報を、使用するネットワークに転送します。

### サポートされているコマンド

オブジェクト :   Get Attribute (01h)  
                   Create (03h) (45 ページの “コマンドの詳細 : Create” を参照)  
                   Delete (04) (45 ページの “コマンドの詳細 : Delete” を参照)

インスタンス :   Get Attribute (01h)

### オブジェクトの属性 (インスタンス #0)

#	名前	アクセス	データ・タイプ	値
1	Name	Get	CHAR の配列	'Diagnostic'
2	Revision	Get	UINT8	01h
3	Number of instances	Get	UINT16	(作成された診断イベントの数によって異なる)
4	Highest instance no.	Get	UINT16	(ネットワーク特有)
11	Max no. of instances	Get	UINT16	作成できるインスタンスの最大数 (ネットワーク特有) <sup>a</sup>

- a. 最大数のインスタンスのうち、モジュールを強制的に 'EXCEPTION' 状態にする重大度レベル 'メジャー、回復不能' のものに対して、常に 1 つのインスタンスが予約されています。特定のネットワークに対して許可されたインスタンスの数に関する情報については、69 ページの 'ネットワーク比較チャート' を参照してください。

### インスタンスの属性 (インスタンス #1)

#	名前	アクセス	タイプ	説明
1	Severity	Get	UINT8	44 ページの “重大度レベル” を参照
2	Event Code	Get	UINT8	44 ページの “イベント・コード” を参照
3	NW specific extension	Get	UINT8 の配列	ネットワーク特有のイベント情報 (任意)

## 重大度レベル

#	重大度	コメント
00h	マイナー、回復可能	-
10h	マイナー、回復不能	回復不能なイベントは削除できません。
20h	メジャー、回復可能	-
30h	メジャー、回復不能	EXCEPTION に状態移行させます。
(その他)	-	(将来の使用のために予約)

## イベント・コード

#	意味	コメント
10h	一般的なエラー	-
20h	電流	-
21h	電流、デバイス入力側	-
22h	電流、デバイスの内側	-
23h	電流、デバイス出力側	-
30h	電圧	-
31h	電源電圧	-
32h	デバイス内の電圧	-
33h	出力電圧	-
40h	温度	-
41h	周囲温度	-
42h	デバイス温度	-
50h	デバイス・ハードウェア	-
60h	デバイス・ソフトウェア	-
61h	内部ソフトウェア	-
62h	ユーザ・ソフトウェア	-
63h	データ・セット	-
70h	追加モジュール	-
80h	モニタリング	-
81h	通信	-
82h	プロトコル・エラー	-
90h	外部エラー	-
F0h	追加機能	-
FFh	NW 特有	定義はネットワーク特有です。詳細については、各 Network Interface Appendix を参照してください。

## コマンドの詳細 : Create

### 詳細

コマンド・コード : 03h

有効 : オブジェクト・インスタンス

### 説明

新しいインスタンスを作成します。この場合、ホスト・アプリケーション内の新しい診断イベントを表します。

- コマンドの詳細

フィールド	内容
CmdExt[0]	重大度
CmdExt[1]	イベント・コード (前のページを参照)
MsgData[0-n]	ネットワーク特有の拡張機能 (任意、定義はネットワーク特有)

- レスポンスの詳細 (成功)

フィールド	内容
MsgData [0...1]	作成されたインスタンスの番号

## コマンドの詳細 : Delete

### 詳細

コマンド・コード : 04h

有効 : オブジェクト・インスタンス

### 説明

既存のインスタンス (以前に作成された診断イベント) を削除します。

**注意 :** 回復不能なイベントを表すインスタンスは削除できません。

- コマンドの詳細

フィールド	内容
CmdExt[0]	削除するインスタンスの番号 (ロー・バイト)
CmdExt[1]	削除するインスタンスの番号 (ハイ・バイト)

- レスポンスの詳細 (エラー)

エラー	内容	コメント
Object Specific Error	MsgData[0] = FFh	-
	MsgData[1] = 01h	エラー・コード (削除されない)。イベントは削除できません。その理由は、イベント自体が回復不能であるからかネットワーク特有の理由のいずれかです。

下記も参照してください。

- 30 ページの “エラー・コード”

## 7.5 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

このオブジェクトは、ネットワークに関する一般的な情報（ネットワーク・タイプ、データ・フォーマットなど）を保持します。ADI をホスト・アプリケーション側からプロセス・データとしてマッピングする場合にも使用されます。

下記も参照してください。

- 54 ページの “アプリケーション・データ・オブジェクト (FEh) ”
- 63 ページの “アプリケーション・オブジェクト (FFh) ”

### サポートされているコマンド

オブジェクト : Get\_Attribute (01h)  
 インスタンス : Get\_Attribute (01h)  
                   Set\_Attribute (02h)  
                   Get\_Enum\_String (06h)  
                   Map\_ADI\_Write\_Area (10h)  
                   Map\_ADI\_Read\_Area (11h)

### オブジェクトの属性 (インスタンス #0)

#	名前	アクセス	データ・タイプ	値
1	Name	Get	CHAR の配列	'Network'
2	Revision	Get	UINT8	02h
3	Number of instances	Get	UINT16	(モジュール・タイプによって異なる)
4	Highest instance no.	Get	UINT16	(モジュール・タイプによって異なる)

## インスタンスの属性 (インスタンス #1)

#	名前	アクセス	カテゴリ	タイプ	説明
1	Network type	Get	拡張	UINT16	(各 Network Interface Appendix を参照)
2	Network type string	Get	-	CHAR の配列	
3	Data format	Get	基本	ENUM	ネットワーク・データ・フォーマット： 値：列挙文字列： 00h：'LSB First' 01h：'MSB First'
4	Parameter data support	Get	拡張 <sup>a</sup>	BOOL	この属性は、ネットワークがアサイクリック・データ・サービスをサポートするかどうかを示します。 値：意味： True：ネットワークはアサイクリック・データのアクセスをサポートします。 False：アサイクリック・データに対するサポートなし
5	Write Process Data size	Get	-	UINT16	現在のライト・プロセス・データのサイズ (バイト) 全ての成功した Map_ADI_Write_Area または Remap_ADI_Write_Area に対して更新されます。
6	Read Process Data size	Get	-	UINT16	現在のリード・プロセス・データのサイズ (バイト) 全ての成功した Map_ADI_Read_Area または Remap_ADI_Read_Area に対して更新されます。
7	Exception Information	Get	-	UINT8	モジュールが EXCEPTION 状態になった場合、ネットワーク特有の追加情報がここに示される可能性があります (各 Network Interface Appendix を参照)。

a. プロセス・データにマッピングする ADI を決定するために使用できます。

## コマンドの詳細 : Map\_ADI\_Write\_Area

### 詳細

コマンド・コード : 10h

有効 : インスタンス

### 説明

このコマンドは、ADI をライト・プロセス・データとしてマッピングします。成功した場合、レスポンス・データには、ライト・プロセス・データのイメージの最初からのマッピングされた ADI のオフセットが含まれます。

**注意 1 :** ADI を 2 回以上マッピングすることは一部のネットワークで受け入れられていないため、2 回以上のマッピング（リード/ライト・プロセス・データへの複数のマッピング、またはリード/ライト・プロセス・データの両方へのマッピング）を行わないことを強く推奨しています。

**注意 2 :** ADI の一部のみをマッピングすることはできません。常に ADI の全てのエレメントをマッピングする必要があります。

**注意 3 :** 特定の Anybus の実装では、ネットワークはランタイム中にプロセス・データを再配置できます。詳細については、54 ページの“アプリケーション・データ・オブジェクト (FEh)”を参照してください。

下記も参照してください。

- 49 ページの“コマンドの詳細 : Map\_ADI\_Read\_Area”
- 63 ページの“アプリケーション・オブジェクト (FFh)”

**重要 :** エラー制御はコマンド・パラメータでのみ行われます。Anybus モジュールは、実際の ADI のアトリビュートのリード結果によって、これらのパラメータの正しさを確認することはしません。

- コマンドの詳細

フィールド	内容
CmdExt[0]	ADI のインスタンス番号 (ロー・バイト)
CmdExt[1]	ADI のインスタンス番号 (ハイ・バイト)
MsgData[0]	ADI のデータ・タイプ (27 ページの“データ・フォーマット”を参照)
MsgData[1]	ADI のエレメントの数
MsgData[2]	ADI のオーダー番号 (ロー・バイト)
MsgData[3]	ADI のオーダー番号 (ハイ・バイト)

**注意 :** マッピング・コマンド内のオーダー番号は、アプリケーション・データ・オブジェクト内の'Get\_Instance\_Number\_By\_Order'コマンドのオーダー番号と等しくなります。

下記も参照してください。

- 54 ページの“アプリケーション・データ・オブジェクト (FEh)” (オーダー番号)

- レスポンスの詳細 (成功)

フィールド	内容
MsgData[0]	ライト・プロセス・データの開始からマッピングされた ADI のオフセット

- レスポンスの詳細 (エラー)

エラー	内容
Invalid CmdExt[0]	ADI の番号は有効ではありません。
Invalid state	ADI のマッピングは、'SETUP' 状態でのみ可能です。
Object Specific Error	オブジェクト特有のエラー (詳細については、MsgData[1] を参照)
01h : 無効なデータ・タイプ	データ・タイプはプロセス・データには有効ではありません。
02h : エレメントの無効な数	エレメントの数は有効ではありません (ゼロ)。
03h : 無効な全体サイズ	結果として発生したデータ・サイズ全体が最大許容サイズ (ネットワーク・タイプによって異なる) を超えるため、要求されたマッピングは拒否されます。
04h : 複数のマッピング	特定のネットワークが ADI の複数のマッピングを受け入れないため、要求されたマッピングは拒否されました。
05h : 無効なオーダー番号	オーダー番号は有効ではありません (ゼロ)。
06h : 無効なマップ・コマンド・シーケンス	コマンド・シーケンスに関するネットワーク特有の要件は、違反されました?

**重要:** エラー制御はコマンド・パラメータでのみ行われます。Anybus モジュールは、実際の ADI のアトリビュートのリード結果によって、これらのパラメータの正しさを確認することはしません。

## コマンドの詳細 : Map\_ADI\_Read\_Area

### 詳細

コマンド・コード : 11h

有効 : インスタンス

### 説明

このコマンドは、ADI をリード・プロセス・データにマッピングする点を除いて、Map\_ADI\_Write\_Area と同じです。

下記も参照してください。

- 48 ページの “コマンドの詳細 : Map\_ADI\_Write\_Area”
- 63 ページの “アプリケーション・オブジェクト (FFh)”

## 7.6 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

ネットワーク特有

### オブジェクトの説明

このオブジェクトには、エンド・ユーザによって設定されるネットワーク特有のコンフィグレーション・パラメータ（一般的に、ボーレートやノード・アドレスなどの設定）が含まれています。このオブジェクト内のインスタンスの実際の定義はネットワーク特有ですが、インスタンス 1 および 2 は、常に 8 ビットのデータ・タイプという点で固定されています。

可能な場合、これらのインスタンスには次の慣例が使用されます。

インスタンス番号	データ・タイプ	パラメータ
1	任意の 8 ビット・データ・タイプ	現在選択されているネットワーク・デバイスのアドレス（または同様のもの）
2	任意の 8 ビット・データ・タイプ	現在選択されているネットワーク通信ビットレート（または同様のもの）

このオブジェクトでのインスタンスの値は、その最初の値が変化するたびに更新する必要があります。従って、機械スイッチまたは同様のものは、ホスト・アプリケーションによって継続的にモニタリングする必要があります。

**注意：**'共有' アクセスがタグ付けされたインスタンス（記述子によって示される）は、**揮発性**としてみなす必要があります。このようなインスタンスに対する 'set' アクセスは、その値を変更する場合と変更しない場合があります。その値が影響を受けない場合は、Anybus モジュールはエラーで応答しません。

### 入力デバイスの細分化

Anybus モジュールは "ハードワイヤード" 入力デバイス（物理的な機械スイッチなど）から発生するパラメータとキーパッドやディスプレイなどの "ソフト" 入力デバイスによって指定されるパラメータを区別します。これによって、Anybus モジュールは、パラメータの実際の入力デバイスに関連したネットワーク特有のニーズを満たすことができます（一部のネットワークでは、物理スイッチでの値の変化はオンボード LED で視覚的に承認される必要があります）。

この区別は、ホスト・アプリケーションからの次のアクションに基づいています（表を参照）。

状態	アクション（ホスト・アプリケーション）	Anybus の挙動
SETUP	物理スイッチから発生するパラメータをポーリングし、更新します（影響を受けたパラメータのそれぞれに対して、少なくとも 1 つの 'set' コマンドを発行するようにする）。"ソフト" 入力デバイスから発生するパラメータは更新しません（これらのパラメータの 'set' コマンドはまだ発行しないこと）。	Anybus モジュールは、影響を受けたパラメータを物理スイッチから発生したものと識別します。残りのパラメータは、"ソフト" 入力デバイスから発生すると想定されています。
（その他の状態）	必要に応じて、全てのパラメータ（物理スイッチと "ソフト" 入力方法）をポーリングし、更新します。	Anybus モジュールは SETUP 状態中に更新されたパラメータの記録を取るため、ネットワークが必要とする場合、パラメータに対して異なる処理を行うことができます。



## サポートされているコマンド

オブジェクト :   Get\_Attribute (01h)  
                   Reset (05h) (実際の挙動はネットワーク特有)

インスタンス :   Get\_Attribute (01h)  
                   Set\_Attribute (02h)  
                   Get\_Enum\_String (06h)

## オブジェクトの属性 (インスタンス #0)

#	名前	アクセス	データ・タイプ	値
1	Name	Get	CHAR の配列	'Network Configuration'
2	Revision	Get	UINT8	01h
3	No. of instances	Get	UINT16	(ネットワークに依存)
4	Highest instance no.	Get	UINT16	

## インスタンスの属性 (インスタンス #1... N)

各インスタンスは、ネットワーク・コンフィグレーション・パラメータを表します。その中の属性は、パラメータの総合的な記述 (名前、データ・タイプなど) を提供します。

インスタンス名と列挙文字列は多言語に対応しています (10 ページの “多言語によるサポート” を参照)。実際の文字列は当然ネットワーク特有ですが、文字の最大数は 13 字に制限されています。

#	名前	アクセス	カテゴリ	タイプ	説明
1	Name	Get	Appl. Spec. <sup>a</sup>	CHAR の配列	パラメータ名 ('ノード・アドレス' など)
2	Data type	Get	Appl. Spec. <sup>a</sup>	UINT8	データ・タイプ (27 ページの “データ・フォーマット” を参照)
3	Number of elements	Get	Appl. Spec. <sup>a</sup>	UINT8	指定されたデータ・タイプのエレメントの数
4	Descriptor	Get	Appl. Spec. <sup>a</sup>	UINT8	パラメータのアクセス権限を指定するビット・フィールド <u>ビット: アクセス</u> b0: 1: Get アクセス b1: 1: Set アクセス b2: 1: 共有アクセス <sup>b</sup>
5	Value	"記述子" によって決定される	Appl. Spec. <sup>a</sup>	"データ・タイプ" によって決定される	実際のパラメータ値。不揮発性メモリに保存されます。

a. この属性のカテゴリは、アプリケーションでの使用方法によって異なります。

b. '共有' アクセスがタグ付けされたインスタンスは、**揮発性**とみなす必要があります。このようなインスタンスに対する 'set' アクセスは、その値を変更する場合と変更しない場合があります。その値が影響を受けない場合は、Anybus モジュールはエラーで応答しません。

**注意:** インスタンス #1 とインスタンス #2 は、アプリケーションに存在する場合、'基本' としてカテゴリ化されます。このオブジェクトの他の全てのインスタンスは、各ネットワーク・アペンディックスでカテゴリ化されます。

## 8. ホスト・アプリケーション・オブジェクト

### 8.1 概要

このグループのオブジェクトは、ホスト・アプリケーション・ソフトウェア内への実装を意図しています。Anybus モジュールは、これらのオブジェクトに対してコマンドを発行し、これらの中の設定とデータにアクセスします。その機能は、オブジェクトを使用する時期と方法を示すためにカテゴリ化されています。

下記も参照してください。

- 27 ページの “データ・フォーマット”
- 38 ページの “Anybus モジュール・オブジェクト”
- 68 ページの “機能のカテゴリ化”

## 8.2 実装のガイドライン

オブジェクトの実装とは、一般的に着信コマンドの構文を解析することと適切なレスポンスを形成することです。これがどのように行われるかについての詳細情報はこのマニュアルの範囲には含まれていませんが、次の基本的な規則に従うことが重要です。

- 実装されたオブジェクトでは、このマニュアルや Network Interface Appendix で指定された全てのオブジェクト・アトリビュート（インスタンス #0）を特性とする必要があります。
- 何らかの理由でコマンドが実行できない場合（特定のオブジェクト、アトリビュート、またはコマンドが実装されていない場合）、問題の原因を示す適切なエラー・コードを使用して応答します。
- アプリケーション・データ・オブジェクトに対するサポートは必須です。
- アプリケーション・オブジェクトに対するサポートは特に推奨されていますが、任意です。
- ネットワーク特有のオブジェクトに対するサポートは任意です。ただし、Anybus モジュールによって提供される標準機能は、ネットワークの機能を特定のあらかじめ定義されたデバイス情報およびサービスの使用に制限しますので注意してください。これらの制限の重要度は異なります。各 Network Interface Appendix を参照してください。この標準機能が不十分である（ベンダ特有の情報または拡張されたネットワーク機能が必要である）場合、ホスト・アプリケーションでネットワーク特有のオブジェクトが実装される可能性があります。
- 起動中、モジュールはネットワーク特有のオブジェクトでアトリビュートの値を取得しようとします。アトリビュートがホスト・アプリケーションで実装されない場合、エラー・メッセージ（06h、"Invalid CmdExt[0]"）で応答します。モジュールはそのデフォルト値を使用します。また、モジュールがネットワーク・アペンドックスにリスト表示されていないアトリビュートの値を取得しようとする場合、エラー・メッセージ（06h、"Invalid CmdExt[0]"）で応答します。
- プロセス・データ再配置に対するサポート（コマンド 'Remap\_ADI\_Write\_Area' および 'Remap\_ADI\_Read\_Area' による）は、Anybus-CompactCom ドライブ・プロファイル製品系列では必須です。これらのコマンドの実装は標準の CompactCom 製品系列では任意ですが、今後、特定のネットワークにより良いネットワーク統合を提供する可能性があります。

下記も参照してください。

- 30 ページの“エラー・コード”

---

**重要：** オブジェクト改定アトリビュートの目的は、ホスト・アプリケーション内でのオブジェクトの実装と Anybus モジュール内でのオブジェクトの実装に互換性があるかどうかを Anybus モジュールが特定できるようにし、必要な場合は異なる実装を使用できるようにすることです。従って、オブジェクト改定アトリビュートが実際の実装を反映し、このマニュアルや Network Interface Appendix のみでの変更された内容に基づいて増加される必要があります。

ご質問がある場合は、HMS のテクニカル・サポート・サービスまでご連絡ください。

## 8.3 アプリケーション・データ・オブジェクト (FEh)

### カテゴリ

基本。このオブジェクトが必須であることに注意してください。

### オブジェクトの説明

このオブジェクト内の各インスタンス（アプリケーション・データ・インスタンスまたは ADI と呼ばれる）は、ネットワーク上で表すデータのブロックと相互に関連しています。ネットワークからこのようなデータにアクセスするたびに、モジュールはこれらの要求を、このオブジェクト（またはその中のインスタンス）に対するオブジェクト要求に変換します。必要に応じて、モジュールはこのオブジェクトにも自発的にアクセスする可能性があります。ネットワーク上での正確な表示はネットワーク特有です（例えば、DeviceNet では、ADI は専用 CIP オブジェクトとして表されるが、PROFIBUS では、ADI はアサイクリック DP-V1 リードおよびライト・サービスを用いてアクセスされる）。

そのインスタンス番号にかかわらず、ネットワークおよび Anybus モジュールが ADI のホスト・アプリケーションを効率的にスキャンできるようにするために、このオブジェクトは追加の 'Get\_Instance\_Number\_By\_Order' コマンドを実装します。このコマンドは、ADI が番号付きリストでソートされていたかのように ADI インスタンス番号を取得し、Anybus モジュールがホスト・アプリケーションで実際に実装されるインスタンスに対してのみ照会できるようにします。オーダー番号は、ADI をプロセス・データにマッピングする場合にも使用されます（48 ページの“コマンドの詳細：Map\_ADI\_Write\_Area”を参照）。

#### 例：

この例では、ホスト・アプリケーションには、インスタンス番号 1、3、および 100 が付いた 4 つの ADI があります。

インスタンス #	実装	オーダー番号
1	あり	1
2	なし	-
3	あり	2
4... 99	なし	-
100	あり	3

この特定の場合では、ホスト・アプリケーションは、オーダー番号 3 の 'Get\_Instance\_Number\_By\_Order' 要求に対して、インスタンス番号 100 で応答します。

#### 重要：

- Anybus モジュールは、要求が '明らかに' 誤っている場合でも、パラメータ要求のエラー制御に対するホスト・アプリケーションの責任を引き継ぎません（例えば、ゼロ・バイト・データを持つ ADI へのライト要求または存在しないアトリビュートにアクセスする試みは、モジュールによって 'フィルタリング・アウト' されません）。
- 一部のネットワークは特定のタイミング要求を課す場合があるため、ホスト・アプリケーションでのレスポンス時間（このオブジェクトに回答する前に、オブジェクトに対する着信要求を処理するために費やされる時間）を考慮に入れる必要があります。該当する場合、特別のタイミング要件などは、各ネットワーク・アペンディックスで指定されています。

## コマンド

オブジェクト :   Get\_Attribute (01h)  
                   Get\_Instance\_Number\_By\_Order (10h)  
                   Get\_Profile\_Instance\_Numbers (11h)<sup>1</sup>  
                   Remap\_ADI\_Write\_Area (13h)<sup>1,2</sup>  
                   Remap\_ADI\_Read\_Area (14h)<sup>1,2</sup>

インスタンス :   Get\_Attribute (01h)  
                   Set\_Attribute (02h)  
                   Get\_Enum\_String (06h)  
                   Get\_Indexed\_Attribute (07h)  
                   Set\_Indexed\_Attribute (08h)  
                   Get\_ADI\_Info (12h)<sup>2,3</sup>

## オブジェクトのアトリビュート (インスタンス #0)

#	名前	アクセス	データ・タイプ	値
1	Name	Get	CHAR の配列	'Application Data'
2	Revision	Get	UINT8	02h
3	Number of instances	Get	UINT16	(アプリケーションによって異なる)
4	Highest instance no.	Get	UINT16	

1. プロファイル・サポートのある製品を使用する場合、実装は必須です。
2. プロセス・データの再配置がサポートされている場合 (オブジェクト rev. 2 以降)、実装は必須です。
3. 実装はオブジェクト rev. 2 以降で必須です。

## インスタンスの属性 (インスタンス #1... N)

#	名前	アクセス	カテゴリ	タイプ	説明
1	Name	Get	ネットワーク 特有	CHAR の配列	ADI の名前 (多言語可)
2	Data type	Get	基本	UINT8	ADI 値のデータ・タイプ (27 ページの “データ・フォーマット” を参照)
3	Number of elements	Get	基本	UINT8	指定されたデータ・タイプのエレメントの数。一部のネットワークでは受け入れられないので、エレメントの数がゼロに設定された ADI を使用しないように強く推奨されます。
4	Descriptor	Get	基本	UINT8	ADI 値のアクセス権限を指定するビット・フィールド <b>ビット：アクセス</b> b0 : 1 : Get アクセス b1 : 1 : Set アクセス b2 : - : (予約、ゼロに設定) b3 <sup>a</sup> : 1 : ライト・プロセス・データとしてマッピングできます。 b4 <sup>a</sup> : 1 : リード・プロセス・データとしてマッピングできます。
5	Value(s) <sup>b</sup>	"記述子" によって決定される	基本	"データ・タイプ" によって決定される	ADI 値
6	Max. value <sup>b,c</sup>	Get	Appl. Spec. <sup>d</sup>		最大許容 ADI 値。この属性の実装は任意です。実装されない場合、モジュールはこの属性に指定されたデータ・タイプの最大値を使用します。
7	Min. value <sup>b,c</sup>	Get	Appl. Spec. <sup>d</sup>		最小許容 ADI 値。この属性の実装は任意です。実装されない場合、モジュールはこの属性に指定されたデータ・タイプの最小値を使用します。
8	Default value <sup>b,c</sup>	Get	Appl. Spec. <sup>d</sup>		デフォルトの ADI 値。この属性の実装は任意です。実装されない場合、ゼロの値 (浮動 : + 最小値) が使用されます。

- プロセス・データの再配置がサポートされている場合 (オブジェクト改定 2 以降) に必須です。
- これらの属性のバイト・オーダーはネットワーク特有です。Anybus はバイト・スワップなどを行いません。
- 最大 / 最小 / デフォルトの属性は、ADI 内の全てのエレメントに共通しています。つまり、配列の各エレメントに対して個別の最大 / 最小 / デフォルトの値はありません。
- この属性のカテゴリは、アプリケーションでの使用方法によって異なります。

### 重要 :

- エレメントの数は一度定義されると固定され、データ・タイプとともに、配列を処理するために必要なバッファ・スペースを表す必要があります。これはランタイム中のエレメントの数に対して許可された変更ではありません。
- インスタンス値は、メッセージ・データ・フィールドに完全に適合する必要があります。従って、エレメントの数に各エレメントのサイズ (バイト) を掛けた値が 255 バイトを超えてはなりません。
- ランタイム中に変更される可能性がある属性は、属性 #1 ('Name') および #5 ('Value') のみです。一度定義されると、他の全ての属性は固定されているとみなす必要があります。ランタイム中にこれらを変更することは許可されていません。

## コマンドの詳細 : Get\_Instance\_Number\_By\_Order

### 詳細

コマンド・コード : 10h

有効 : オブジェクト

### 説明

このコマンドは、あたかもオーダーリストでソートされていたかのように ADI の実際のインスタンス番号を要求します。

- コマンドの詳細 :

フィールド	意味
CmdExt[0]	要求されたオーダー番号 (ロー・バイト)
CmdExt[1]	要求されたオーダー番号 (ハイ・バイト)

- レスポンスの詳細 (成功) :

フィールド	意味
MsgData [0...1]	プロファイル ADI のインスタンス番号

- レスポンスの詳細 (エラー) :

エラー・コード	意味
Invalid CmdExt[0]	要求されたオーダー番号は、ADI に関連付けられていません。

下記も参照してください。

- 54 ページの “オブジェクトの説明”

### 8.3.1 コマンドの詳細 : Get\_Profile\_Instance\_Numbers

#### 詳細

コマンド・コード : 11h

有効 : オブジェクト

#### 説明

**注意 :** このコマンドは、Anybus-CompactCom ドライブ・プロファイル製品系列など、ネットワーク・プロファイルのビルトイン・サポートがある製品にのみ該当します。標準バージョンの Anybus-CompactCom では使用されません。

このコマンドは、特定のプロファイルに関連付けられた ADI インスタンス番号のリストを取得します。このために使用される ADI の数、それらの目的、データ・タイプなどは、実際のプロファイルによって規定されています。ただし、アプリケーションは必要に応じてインスタンス番号を実装に合わせて割り当てる場合があります。

**重要 :** Anybus-CompactCom ドライブ・プロファイル製品系列を使用する場合、このコマンドの実装は必須です。詳細については、"Anybus-CompactCom Drive Profile Design Appendix" を参照してください。標準バージョンの Anybus-CompactCom を使用する場合、このコマンドの実装は不要です。

- コマンドの詳細 :

フィールド	意味
CmdExt[0]	(予約、無視する)
CmdExt[1]	値 : プロファイル 00h : (予約) 01h : ドライブ・プロファイル (その他) (将来の使用のために予約)

- レスポンスの詳細 :

フィールド	タイプ	意味
MsgData [0..1]	UINT16	プロファイル ADI 1 のインスタンス番号
MsgData [2..0.3]	UINT16	プロファイル ADI 2 のインスタンス番号
...	...	...
MsgData [(2n-2)..(2n-1)]	UINT16	プロファイル ADI n のインスタンス番号

**注意 1 :** このリストの内容は完全にプロファイルに依存しており、各プロファイル・アペンディックス (Anybus-CompactCom Drive Profile Design Appendix) で指定されています。

**注意 2 :** ホスト・アプリケーションによってサポートされていない任意または条件付きプロファイル ADI には、ゼロのインスタンス番号が示されます。

**注意 3 :** Anybus モジュールは、次のいずれかが発生した場合、'EXCEPTION' 状態になります。

- レスポンスのサイズが要求されたプロファイルと一致しなかった。
- 要求されたパラメータがサポートされていないとしてマーキングされた (注意 2 を参照)。
- アプリケーションがレスポンスのエラーを示した。

下記も参照してください。

- 54 ページの “オブジェクトの説明”



### 8.3.2 コマンドの詳細 : Get\_ADI\_Info

#### 詳細

コマンド・コード : 12h

有効 : インスタンス

#### 説明

このコマンドは、オブジェクト・アトリビュートであるデータ・タイプ、エレメントの数、および ADI の記述子を 1 つのレスポンス・メッセージに収集するために使用されます。

**注意 :** 実装はオブジェクト rev. 2 以降で必須です。

- コマンドの詳細 :

フィールド	意味
CmdExt[0]	(予約、無視する)
CmdExt[1]	

- レスポンスの詳細 (成功) :

フィールド	意味
MsgData[0]	データ・タイプ
MsgData[1]	エレメントの数
MsgData[2]	記述子

- レスポンスの詳細 (エラー) :

エラー・コード	意味
04h	サポートされていないインスタンス

下記も参照してください。

- 54 ページの “アプリケーション・データ・オブジェクト (FEh) ”

### 8.3.3 コマンドの詳細 : Remap\_ADI\_Write\_Area

#### 詳細

コマンド・コード : 13h

有効 : オブジェクト

#### 説明

ネットワークがプロセス・データ・マップ内の変更を要求すると、Anybus モジュールはこのコマンドを発行します。ADI はコマンドによる記述と同じ順序で挿入ポイントでマッピングされます。コマンドは複数のマッピング項目を削除 / 挿入し、CmdExt[0] 内のマッピング項目番号によって示されるポイントで開始できます。この場合のマッピング項目は、Map\_ADI\_Write\_Area コマンドによって以前マッピングされた ADI、または Remap\_ADI\_Write\_Area コマンドによって以前マッピングされた ADI (またはマルチエレメント ADI のエレメント) です。

次のデータのセットは、挿入された各マッピング項目のコマンド・データに含まれていません。

- ADI 番号
- 最初にマッピングするエレメントに対するインデックス
- マッピングする連続エレメントの数

コマンドは、SETUP および EXCEPTION を除く全ての状態で発行される可能性があります。

**注意 1 :** コマンドで指定された全てのアクションは、実行されるか拒否されるかのいずれかです。コマンドが受け入れられなかった場合、プロセス・データ・マップは変更されないものとします。

**注意 2 :** Anybus モジュールは、1 回に 1 つの顕著な再配置コマンドに制限されています。

下記も参照してください。

- 46 ページの “ネットワーク・オブジェクト (03h) ”
- 77 ページの “プロセス・データのランタイム再配置 ”
- 81 ページの “例 : Remap\_ADI\_Write\_Area”

---

**重要 :** この手順をサポートするには、ホスト・アプリケーションは、ランタイム中にプロセス・データ・マップを再配置できる必要があります。これはオブジェクト rev. 2 以降で必須の要件です。このオブジェクト改定のサポートは、Anybus-CompactCom ドライブ・プロファイル製品系列を使用する場合には必要ですが、標準製品系列の特定のネットワークにも、より良いネットワーク統合を提供することができます。

• コマンドの詳細 :

フィールド	意味
CmdExt[0]	再配置の開始 (マッピング項目番号、0 = 最初)
CmdExt[1]	(予約、無視する)
Data[0-1]	削除する現在のマッピング項目の数 (0... 256)
Data[2-3]	挿入するマッピング項目の数 (0... 62)
Data[4-5]	新しいマッピング項目 1 : ADI 番号
Data[6]	新しいマッピング項目 1 : 最初にマッピングするエレメントに対するインデックス
Data[7]	新しいマッピング項目 1 : マッピングする連続エレメントの数
Data[8-9]	新しいマッピング項目 2 : ADI 番号
Data[10]	新しいマッピング項目 2 : 最初にマッピングするエレメントに対するインデックス
Data[11]	新しいマッピング項目 2 : マッピングする連続エレメントの数
...	(その他)

• レスポンスの詳細 (成功) :

フィールド	意味
MsgData[0]	結果として生じるライト・プロセス・データ領域の合計サイズ (バイト) (ロー・バイト)
MsgData[1]	結果として生じるライト・プロセス・データ領域の合計サイズ (バイト) (ハイ・バイト)

• レスポンスの詳細 (エラー) :

エラー・コード	エラー	意味
01h	マッピング項目エラー	要求されたマッピングは、少なくとも1つのマッピング項目に対する NAK のために拒否されます。
02h	無効な全体サイズ	結果として発生したデータ・サイズ全体がアプリケーションの最大許容サイズを超えるため、要求されたマッピングは拒否されます。

### 8.3.4 コマンドの詳細 : Remap\_ADI\_Read\_Area

#### 詳細

コマンド・コード : 14h

有効 : オブジェクト

#### 説明

このコマンドは、ADI をリード・プロセス・データ領域に（再）マッピングするために使用されます。それ以外は Remap\_ADI\_Write\_Area と同等です。

**注意 1 :** ACK の再配置コマンドへの移行の成功は、プロセス・データ・マップが変更されるポイントを示します。シリアル・アプリケーションでは、これは変更されたプロセス・データ・マップが ACK の後の空のテレグラムに続くテレグラム（または再送信の場合のテレグラム）で予想されているか使用されることを意味します（77 ページの“プロセス・データのランタイム再配置”を参照）。

**注意 2 :** 再配置中の変更されたデータ指示（予備ビット）の処理については、15 ページの“変更データ表示”で説明しています。

下記も参照してください。

- 46 ページの“ネットワーク・オブジェクト (03h)”
- 77 ページの“プロセス・データのランタイム再配置”
- 81 ページの“例 : Remap\_ADI\_Write\_Area”

---

**重要 :** この手順をサポートするには、ホスト・アプリケーションは、ランタイム中にプロセス・データ・マップを再配置できる必要があります。これは Anybus-CompactCom ドライブ・プロファイル製品系列の場合には必須の要件ですが、標準製品系列の特定ネットワークにも、より良いネットワーク統合を提供することができます（これらの製品では、このコマンドに対するサポートは特に推奨されていますが、任意です）。

## 8.4 アプリケーション・オブジェクト (FFh)

### 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一般的な設定をグループ化します。このオブジェクトとそのコマンドを実装し、複数の言語とネットワーク・リセット要求をサポートできるようにすることは必須ではありませんが、特に推奨されています。

### サポートされているコマンド

オブジェクト :   Get\_Attribute (01h)  
                   Reset (05h)  
                   Reset\_Request (10h)  
                   Change\_Language\_Request (11h)

インスタンス :   Get\_Attribute (01h)  
                   Set\_Attribute (02h)  
                   Get\_Enum\_String (06h)

## オブジェクトの属性 (インスタンス #0)

#	名前	アクセス	データ・タイプ	値
1	Name	Get	CHAR の配列	'Application'
2	Revision	Get	UINT8	01h
3	Number of instances	Get	UINT16	0001h
4	Highest instance no.	Get	UINT16	0001h

## インスタンスの属性 (インスタンス #1)

#	名前	アクセス	カテゴリ	タイプ	説明
1	Configured	Get	拡張	BOOL	<p>アプリケーションのパラメータが、すぐに使える値から変更されたかどうかを示します。  <u>値：意味：</u>                      False：工場出荷時設定にリセット                      True：設定済み、設定が変更された</p> <p>下記も参照してください。                      - 65 ページの “コマンドの詳細：Reset”                      - 66 ページの “コマンドの詳細：Reset_Request”</p>
2	Supported languages	Get	拡張	ENUM の配列	<p>ホスト・アプリケーションによってサポートされている言語を指定するリスト  <u>値：列挙文字列：</u>                      00h: 'English'                      01h: 'Deutch'                      02h: 'Español'                      03h: 'Italiano'                      04h: 'Français'</p> <p>下記も参照してください。                      - 10 ページの “多言語によるサポート”                      - 39 ページの “Anybus オブジェクト (01h)” (インスタンス・属性 #9)                      - 67 ページの “コマンドの詳細：Change_Language_Request”</p>

## コマンドの詳細 : Reset

### 詳細

コマンド・コード : 05h

有効 : オブジェクト・インスタンス

### 説明

このコマンドは、リセットが要求された場合、モジュールによって発行されます。ネットワーク・タイプによって、'Reset\_Request' コマンドが先行する場合としない場合があります。

- コマンドの詳細

フィールド	内容	コメント
CmdExt[0]	(予約、無視する)	-
CmdExt[1]	00h : 電源オン・リセット	これはデバイスのリセットとしてみなされます。ホスト・アプリケーションは、/RESET 信号を経由してモジュールをリセットします。  <b>注意</b> : Anybus モジュールは、このタイプの要求を発行する前に 'EXCEPTION' 状態になります。
	01h : 工場出荷時設定にリセット	これはホスト・アプリケーションを工場出荷時設定に戻します。モジュールをこの状態に設定するために必要なネットワーク特有の手順は、自動的に行われます。
	02h : 電源オン + 工場デフォルト	上記の 2 つの組み合わせ  <b>注意</b> : Anybus モジュールは、このタイプの要求を発行する前に 'EXCEPTION' 状態になります。

- レスポンスの詳細  
(データなし)

下記も参照してください。

- 64 ページの “インスタンスのアトリビュート (インスタンス #1) ” (アトリビュート #1、'Configured')
- 66 ページの “コマンドの詳細 : Reset\_Request”

## コマンドの詳細 : Reset\_Request

### 詳細

コマンド・コード : 10h

有効 : オブジェクト・インスタンス

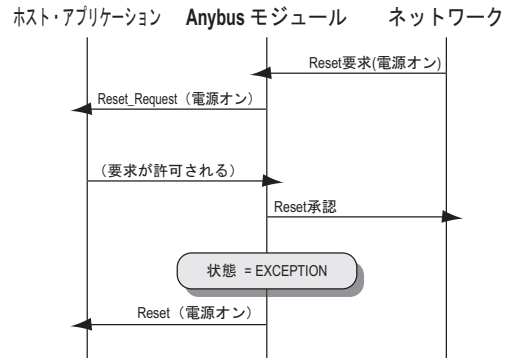
### 説明

特定のネットワークでは、このコマンドは Reset コマンドの前に発行される場合があります (下記を参照)。これは、名前が暗に示すように要求であり、実際のリセット・コマンドではありません。

要求されるリセットは、電源オン・リセット、工場出荷時設定にリセット、またはその両方です。電源オン・リセットは、デバイスのリセットとしてみなされます。

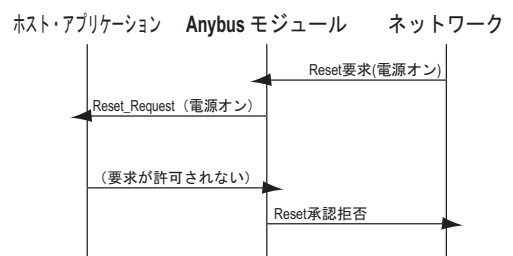
要求が許可された場合、ホスト・アプリケーションも対応する Reset コマンドを受信する準備ができています (図を参照)。

ホスト・アプリケーションは、リセットが何らかの理由で実行できない場合も、自由にエラーで応答できます。このような場合、Reset コマンドはモジュールによって発行されません。



- コマンドの詳細

フィールド	内容
CmdExt[0]	(予約、無視する)
CmdExt[1]	00h = 電源オン・リセット
	01h = 工場出荷時設定にリセット
	02h = 電源オン+工場デフォルト



- レスポンスの詳細

(データなし)

下記も参照してください。

- 64 ページの “インスタンスのアトリビュート (インスタンス #1) ” (アトリビュート #1, 'Configured')
- 65 ページの “コマンドの詳細 : Reset”



## コマンドの詳細 : Change\_Language\_Request

### 詳細

コマンド・コード : 11h

有効 : オブジェクト・インスタンス

### 説明

このコマンドは、現在の言語の変更がネットワークから要求された場合、モジュールによって発行されます。

受け入れられた場合、Anybus オブジェクト (01h) 内で言語アトリビュート (#9) の対応する変更が行われます。ホスト・アプリケーションは、それに応じて、その内部言語設定も調整する必要があります。

- コマンドの詳細

フィールド	内容
CmdExt[0]	(予約、無視する)
CmdExt[1]	要求される言語は次のとおりです。 値 : 言語 : 00h: English 01h: German 02h: Spanish 03h: Italian 04h: French

- レスポンスの詳細

(データなし)

下記も参照してください。

- 10 ページの “多言語によるサポート”
- 39 ページの “Anybus オブジェクト (01h) ” (インスタンス・アトリビュート #9、'Language')
- 64 ページの “インスタンスのアトリビュート (インスタンス #1) ” (アトリビュート #2、'Supported languages')

## A. 機能のカテゴリ化

Anybus CompactCom とアプリケーションのオブジェクト（アトリビュートとサービスを含む）は、3つのカテゴリ（基本、高度、および拡張）に分類されます。

### A.1 基本

このカテゴリには、実装時または使用時に必要なオブジェクト、アトリビュート、およびサービスが含まれます。これらは Anybus CompactCom を起動して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十分な情報です。産業用ネットワークの基本的な機能が使用されます。

これらのオブジェクトについては、このマニュアル（および適切な産業用ネットワークのアペンディックス）で説明しています。製品を認証できるようにする追加オブジェクトなどもこのカテゴリに属しています。

### A.2 拡張

このカテゴリにあるオブジェクトを使用すると、アプリケーションの機能が拡張されます。データの基本的な移動だけでなく、産業用ネットワークのより具体的な特徴にアクセスできます。アプリケーションには付加価値が与えられます。

### A.3 高度

このグループに属するオブジェクト、アトリビュート、およびサービスは、特殊な使用頻度の低い機能を提供します。使用できるネットワーク機能のほとんどは有効で、アクセス可能です。通常、産業用ネットワークの仕様へのアクセスが必要です。

## B. ネットワーク比較チャート

Anybus-CompactCom ソフトウェア・インターフェースは、ネットワーク機能やホスト・システムとの統合を損なうことなく、可能な限り包括的に設計されています。

ホスト・アプリケーションを設計する場合、各ネットワーキング・システムの制限事項と可能性について認識することが重要です。ほとんどの場合、特定のネットワークをサポートするための、追加のソフトウェア・サポートは必要ありません。ただし、ネットワーク機能の特定の側面を十分に活用するには、ある程度の専用ソフトウェア・サポートが必要になる場合があります。

異なるネットワーク実装によって提供される機能の要約を次のページの表に示します。

### 表の解釈のについて：

- 数値は、一般的な実装で予想される値を示します。
- 括弧内の数値は、専用ソフトウェア・サポートを使用した場合に可能な値を示します。
- 最大数の診断インスタンスのうち、モジュールを強制的に 'EXCEPTION' 状態にする重大度レベル 'メジャー、回復不能' に対して、常に 1 つのインスタンスが予約されています。
- データ・タイプがサポートされていない場合、その特定のタイプに完全に対応するものがネットワークにないことを意味します。しかし、そのデータは、他のフォーマットではありますが、ネットワーク上で表示されている場合があります (例えば、UINT64 は 4 つの UINT16 として表される場合がある)。

**注意：**この章の情報は、異なるネットワーク実装での可能性について大まかに示しています。特定のネットワークについての詳細情報は、対応するネットワーク・アペンディックスを参照してください。

項目	EtherNet/IP、1ポート および2ポート	CC-Link	PROFIBUS DP-V1	PROFIBUS DP-V0	PROFINET、1ポート および2ポート	DeviceNet	CA Nopen	Modbus RTU	Modbus/TCP	ControlNet	EtherCAT	CompoNet	SERCOS III
ネットワーク・データ・フォーマット	LSB が先	LSB が先	MSB が先	MSB が先	MSB が先	LSB が先	LSB が先	LSB が先	LSB が先	LSB が先	LSB が先	LSB が先	LSB が先
アサイクリック・データのサポート	あり	なし	あり	なし	あり	あり	あり	あり	あり	あり	あり	あり	あり
1つの ADI でのエレメントの最大数	255	255	64 (240)	該当なし	255	255	254	32 (255)	32 (255)	255	255	255	255
最大 ADI サイズ (バイト)	255	255	64 (240)	該当なし	255	255	254	32 (255)	32 (255)	255	255	255	255
アドレス指定可能な最小 ADI 番号	1	該当なし	1	該当なし	1	1	1	1	1	1	1	1	1
アドレス指定可能な最大 ADI 番号	65535	該当なし	65025	該当なし	32767	65535	16383	4062 (65023)	4062 (65023)	65535	16383	255 <sup>a</sup>	32767 <sup>b</sup>
最大ライト・プロセス・データ (バイト)	256	14.44 (256) <sup>c</sup>	152 (244) <sup>d</sup>	244	256 <sup>d</sup>	256	256	256	256	256	256	32	256
最小ライト・プロセス・データ (バイト)	0	0	0	0	0	0	0	0	0	0	0	0	0
最大リード・プロセス・データ (バイト)	256	14.46 (256) <sup>c</sup>	152 (244)	244	256	256	256	256	256	256	256	32	256
最小リード・プロセス・データ (バイト)	0	0	0	0	0	0	0	0	0	0	0	0	0
最大プロセス・データ (リード+ライト、バイト)	512	28.90 (512) <sup>c</sup>	152 (400) <sup>d</sup>	380	512 <sup>d</sup>	512	512	512	512	512	512	64	512
最小プロセス・データ (リード+ライト、バイト)	0	0	1	1	1	0	0	0	0	0	0	1	0
'Get/Set_Indexed_Attribute' を要求	なし	なし	なし	なし	なし	なし	あり	なし	なし (あり)	なし	あり	なし	なし
'Get_Instance_Number_By_Order' を要求	なし (あり)	なし	なし	なし	なし	なし (あり)	なし	なし	なし (あり)	なし (あり)	あり	なし (あり)	あり
診断インスタンスの最大数	6	6	6 (1)	10	6 (1)	6	6	6	6	6	6	6	6
ネットワーク・リセット・タイプ 0: '電源オン・リセット' をサポート	あり	なし	なし	なし	なし	あり	あり	なし	なし	あり	なし	あり	なし
ネットワーク・リセット・タイプ 1: '工場出荷時設定にリセット' をサポート	あり	なし	なし	なし	あり	あり	あり	なし	なし	あり	あり	あり	あり
SINT64 をサポート	あり	あり	あり	あり	あり	あり	あり	あり	あり	あり	あり	あり	あり
UINT64 をサポート	あり	あり	あり	あり	あり	あり	あり	あり	あり	あり	あり	あり	あり
FLOAT をサポート	あり	あり	あり	あり	あり	あり	あり	あり	あり	あり	あり	あり	あり

- さらに多くの ADI を作成できますが、CompoNet からは 255 個のみをアドレス指定できます。
- これらのうち、約 16000 の ADI に SERCOS III から到達できます。
- プロセス・データの量は、マッピングされた ADI のデータ・タイプによって異なります。詳細については、CC-Link の Network Interface Appendix を参照してください。
- アドレス指定の制限事項については、PROFIBUS および PROFINET の Network Interface Appendix を確認します。

## C. タイミングと性能

### C.1 概要

この章では、Anybus-CompactCom ファミリの各メンバーのために検証され、文書化されたタイミングと性能のパラメータを指定します。

次のタイミングの各側面が測定されます。

カテゴリ	パラメータ	ページ
起動の遅延	T1、T2	72 ページ
NW_INIT 遅延	T3	72 ページ
テレグラム遅延	T4	73 ページ
コマンド遅延	T5	74 ページ
Anybus リード・プロセス・データ遅延 (Anybus 遅延)	T6、T7、T8	75 ページ
Anybus ライト・プロセス・データ遅延 (Anybus 遅延)	T12、T13、T14	76 ページ
ネットワーク・システム・リード・プロセス・データ遅延 (ネットワーク・システム遅延)	T9、T10、T11	76 ページ
ネットワーク・システム・ライト・プロセス・データ遅延 (ネットワーク・システム遅延)	T15、T16、T17	76 ページ

**重要：** ライトした時点では、全てのネットワークのネットワーク特有のタイミング仕様は、まだ公式にリリースされていません。この情報は、使用できる場合、全ての *Network Interface Appendix* に継続的に追加されます。

ご質問がある場合は、HMS までご連絡ください。

## C.2 内部タイミング

### C.2.1 起動の遅延

次のパラメータは、/RESET がリリースされるポイントから指定されたイベントが発生するポイントまで測定された時間として定義されます。

パラメータ	説明	最大	単位
T1	Anybus は最初のアプリケーションの割り込みを生成します (パラレル・モード)。	1.0	s
T2	Anybus は最初のアプリケーションのテレグラムを受信し、処理できません (シリアル・モード)。	1.0	s

### C.2.2 NW\_INIT 遅延

NW\_INIT 状態で必要なアクションを行うために Anybus モジュールによって要求される時間は、ネットワーク固有です。さらに、この状態でホスト・アプリケーションに対して発行されるコマンドの数は、異なるネットワーク間だけでなく異なる実装間においても (実際のプロセス・データの実装などによって) 変化する可能性があります。これは、ホスト・アプリケーションのレスポンス時間がこのパラメータにも大きな影響を与えることを意味しています。従って、一般的なホスト・アプリケーションの実装とともに、全ての Anybus バージョンが満たすことのできる最大値を指定することのみ可能です。

このパラメータを決定することは、その妥当性についてホスト・アプリケーションが監視することを要求または期待するものではありません。Anybus モジュールの健全さは、プロトコルが動作していること、および正しい状態が表示されているということにより、十分に示されます。ただし、Anybus のコンセプトがこの点で信頼されていない場合、ホスト・アプリケーションは、機能していない状況がエンド・ユーザに示される前に、タイムアウトを待つ場合があります。この場合は、起動段階中であるため、やや長いタイムアウト値を使用するほうが良いでしょう。

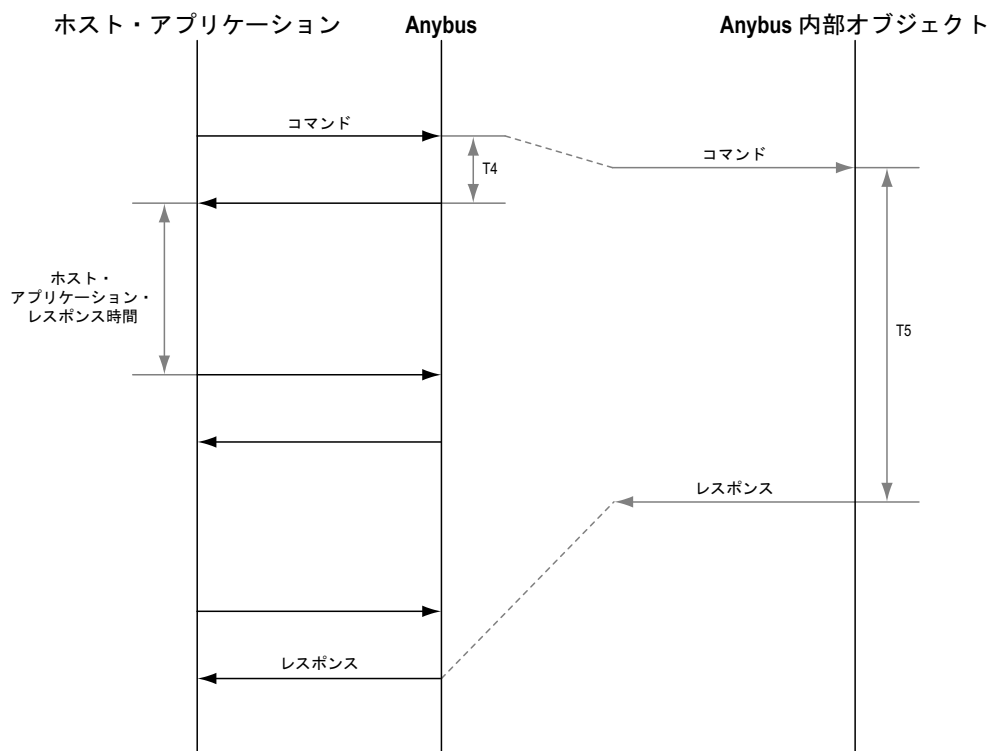
パラメータ	条件
ネットワーク特有のコマンドの数	最大
各方向のプロセス・データにマッピングされる ADI (1つの UINT8) の数	32 <sup>a</sup>
アプリケーション・レスポンス時間	< 10 ms
アプリケーションが同時に処理できる未解決 Anybus コマンド数	1

a. または、最大数 (ネットワーク特有の最大値がより小さい場合)

パラメータ	説明	通信	最大	単位
T3	NW_INIT 遅延	シリアル 19.2 kbps	30	s
		(他の全てのモード)	10	s

## C.3 Anybus のレスポンス時間

### C.3.1 概要



### C.3.2 テレグラム遅延

テレグラム遅延は、テレグラムに応答するために Anybus モジュールによって要求される時間として定義されます。

コマンドは1つずつ発行され、以前のコマンドへのレスポンスを受信する前に、Anybus モジュールに対して新しいコマンドが発行されないものと想定されています。

パラメータ	条件
通信	全てのモード
ホスト・アプリケーション・レスポンス時間	$\geq 0.2$ ms
Anybus の状態	全ての状態
各方向のプロセス・データにマッピングされる ADI (1つの UINT8) の数	$32^a$
バスの負荷、ノードの数、ボーレートなど	通常
同時未解決アプリケーション・コマンドの数	1

a. または、最大数 (ネットワーク特有の最大値がより小さい場合)

パラメータ	説明	平均	最大	単位
T4	Anybus テレグラム遅延	< 0.4	1.5	ms

### C.3.3 コマンド遅延

コマンド遅延は、内部で処理されるコマンド（ネットワーク情報が交換されないコマンド）に応答するために Anybus モジュールによって要求される時間として定義されます。Anybus モジュールが処理を終了し、レスポンスを送信する準備ができると、測定が終了します。

コマンドは1つずつ発行され、以前のコマンドへのレスポンスを受信する前に、Anybus モジュールに対して新しいコマンドが発行されないと想定されています。

条件	条件
通信	全てのモード
アプリケーション・レスポンス時間	≥0.2 ms
Anybus の状態	全ての状態
各方向のプロセス・データにマッピングされる ADI (1つの UINT8) の数	32 <sup>a</sup>
バスの負荷、ノードの数、ボーレートなど	通常
同時未解決アプリケーション・コマンドの数	1

- a. または、最大数（ネットワーク特有の最大値がより小さい場合）

特定のコマンドでは、様々な技術的な理由のために（不揮発性メモリへのパラメータの保存やファイル・システムのフォーマッティングなど）、かなり多くの実行時間が必要になる場合があります。従って、コマンドはその予測されるコマンド遅延によってカテゴリ化されます。

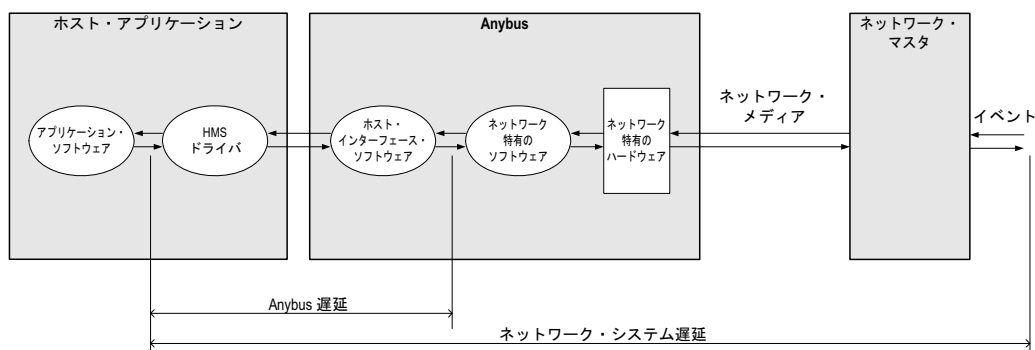
パラメータ	説明	カテゴリ <sup>a</sup>	平均	最大	単位
T5	Anybus コマンド遅延	A	< 1	10	ms
		B	-	5000	ms
		C <sup>b</sup>	-	∞	ms

- a. コマンドは、特に指示がないかぎり、カテゴリ A です。
- b. これらのコマンドは、サービスをブロックするために使用されます。レスポンスは Anybus モジュールが制御しない外部イベントが発生するまで返送されません。ただし、このようなサービスが Anybus モジュールとホスト・アプリケーションの両方で予測できない期間メッセージ・リソースをロックすることは、考慮する必要があります。



## C.4 プロセス・データ

### C.4.1 概要



### C.4.2 Anybus リード・プロセス・データ遅延 (Anybus 遅延)

リード・プロセス・データ遅延（上図のラベルでは 'Anybus 遅延' と表記）は、新しいデータがバッファに入れられ、Anybus ホスト・インターフェース・ソフトウェアで使用できるようになる直前から、データがホスト・アプリケーションで使用できるようになるまで（新しいデータがドライバからリードされた直後）に測定された時間として定義されます。

**注意：**シリアル通信の送信遅延は、これらの測定では考慮されません。

パラメータ	条件
アプリケーション CPU	-
タイマ・システム・コールの間隔	1 ms
ドライバ・コールの間隔	0.2... 0.3 ms
各方向のプロセス・データにマッピングされる ADI (1つの UINT8) の数	8、16、および 32
通信	パラレル
測定期間中のテレグラム・タイプ	プロセス・データのみ
バスの負荷、ノードの数、ボーレートなど	通常

パラメータ	説明	平均	最大	単位
T6	Anybus リード・プロセス・データ遅延、8つの ADI (1つの UINT8)	< 0.5	1	ms
T7	Anybus リード・プロセス・データ遅延、16の ADI (1つの UINT8)	< 0.7	1.2	ms
T8	Anybus リード・プロセス・データ遅延、32の ADI (1つの UINT8)	< 1	1.5	ms

### C.4.3 Anybus ライト・プロセス・データ遅延 (Anybus 遅延)

ライト・プロセス・データ遅延 (図中のラベルでは 'Anybus 遅延' と表記) は、データがホスト・アプリケーションから使用できるようになる時点 (データがホスト・アプリケーションからドライブにライトされる直前) から、新しいデータが Anybus ホスト・インターフェース・ソフトウェアによってネットワーク・バッファに転送された時点までに測定された時間として定義されます。

条件: C-75 “Anybus リード・プロセス・データ遅延 (Anybus 遅延)” と同様。

注意: シリアル通信の送信遅延は、これらの測定では考慮されません。

パラメータ	説明	平均	最大	単位
T12	Anybus ライト・プロセス・データ遅延、8 つの ADI (1 つの UINT8)	< 0.5	1	ms
T13	Anybus ライト・プロセス・データ遅延、16 の ADI (1 つの UINT8)	< 0.7	1.2	ms
T14	Anybus ライト・プロセス・データ遅延、32 の ADI (1 つの UINT8)	< 1	1.5	ms

### C.4.4 ネットワーク・システム・リード・プロセス・データ遅延 (ネットワーク・システム遅延)

ネットワーク・システム・リード・プロセス・データ遅延 (図中のラベルでは 'ネットワーク・システム遅延' と表記) は、ネットワーク・マスターでイベントが生成される時点から、対応するデータがホスト・アプリケーションで使用できるようになる時点 (対応するデータがドライブからリードされた直後) までに測定された時間として定義されます。

パラメータ	説明	平均	最大	単位
T9	ネットワーク・システム・リード・プロセス・データ遅延、8 つの ADI (1 つの UINT8)	(ネットワーク・タイプに依存、各 Network Interface Appendix を参照)		
T10	ネットワーク・システム・リード・プロセス・データ遅延、16 の ADI (1 つの UINT8)			
T11	ネットワーク・システム・リード・プロセス・データ遅延、32 の ADI (1 つの UINT8)			

条件: C-75 “Anybus リード・プロセス・データ遅延 (Anybus 遅延)” と同様。

**重要:** ライトした時点では、全てのネットワークのネットワーク特有のタイミング仕様は、まだ公式にリリースされていません。この情報は、使用できる場合、全ての Network Interface Appendix に継続的に追加されます。

### C.4.5 ネットワーク・システム・ライト・プロセス・データ遅延 (ネットワーク・システム遅延)

ネットワーク・システム・ライト・プロセス・データ遅延 (図で 'ネットワーク・システム遅延' というラベルあり) は、新しいデータがホスト・アプリケーションから使用できるようになる時点 (データがドライブにライトされる直前) から、このデータがネットワーク・マスターで対応するイベント生成する時点まで測定された時間として定義されます。

パラメータ	説明	平均	最大	単位
T15	ネットワーク・システム・ライト・プロセス・データ遅延、8 つの ADI (1 つの UINT8)	(ネットワーク・タイプに依存、各 Network Interface Appendix を参照)		
T16	ネットワーク・システム・ライト・プロセス・データ遅延、16 の ADI (1 つの UINT8)			
T17	ネットワーク・システム・ライト・プロセス・データ遅延、32 の ADI (1 つの UINT8)			

条件: 75 ページの “Anybus リード・プロセス・データ遅延 (Anybus 遅延)” と同様。

**重要:** ライトした時点では、全てのネットワークのネットワーク特有のタイミング仕様は、まだ公式にリリースされていません。この情報は、使用できる場合、全ての Network Interface Appendix に継続的に追加されます。

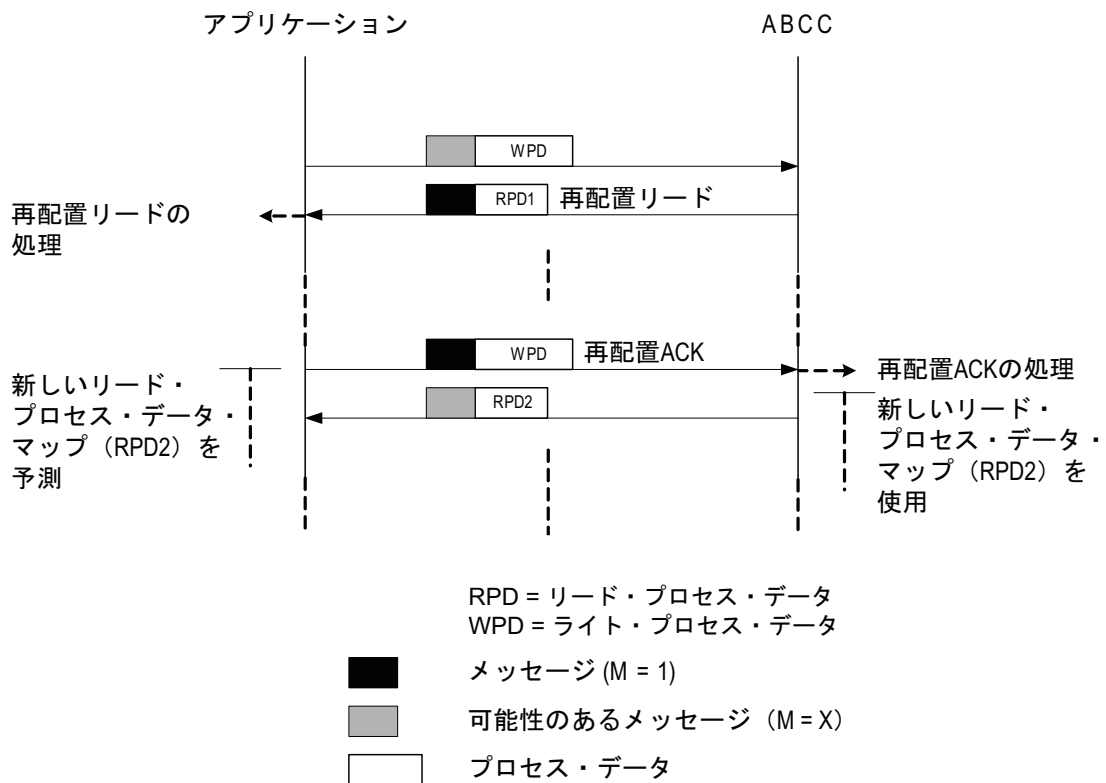
## D. プロセス・データのランタイム再配置

このアペンディックスでは、アプリケーションからの要求を処理し、リードまたはライト・プロセス・データをパラレル・モードおよびシリアル・モードで再配置する方法について説明します。テレグラムがピンポン方式で交換されることに注意してください。

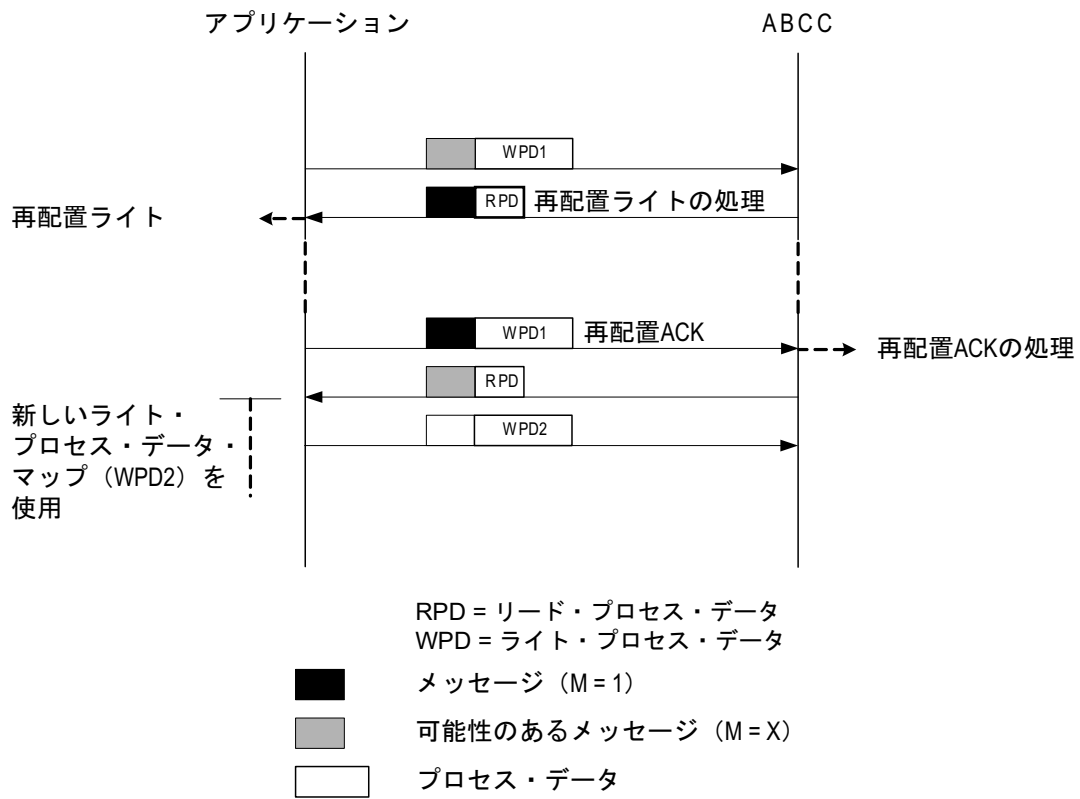
### D.1 パラレル・モード

パラレル・モードでのプロセス・データのランタイム再配置は割合単純です。下の図を参照してください。

#### D.1.1 リード・プロセス・データ



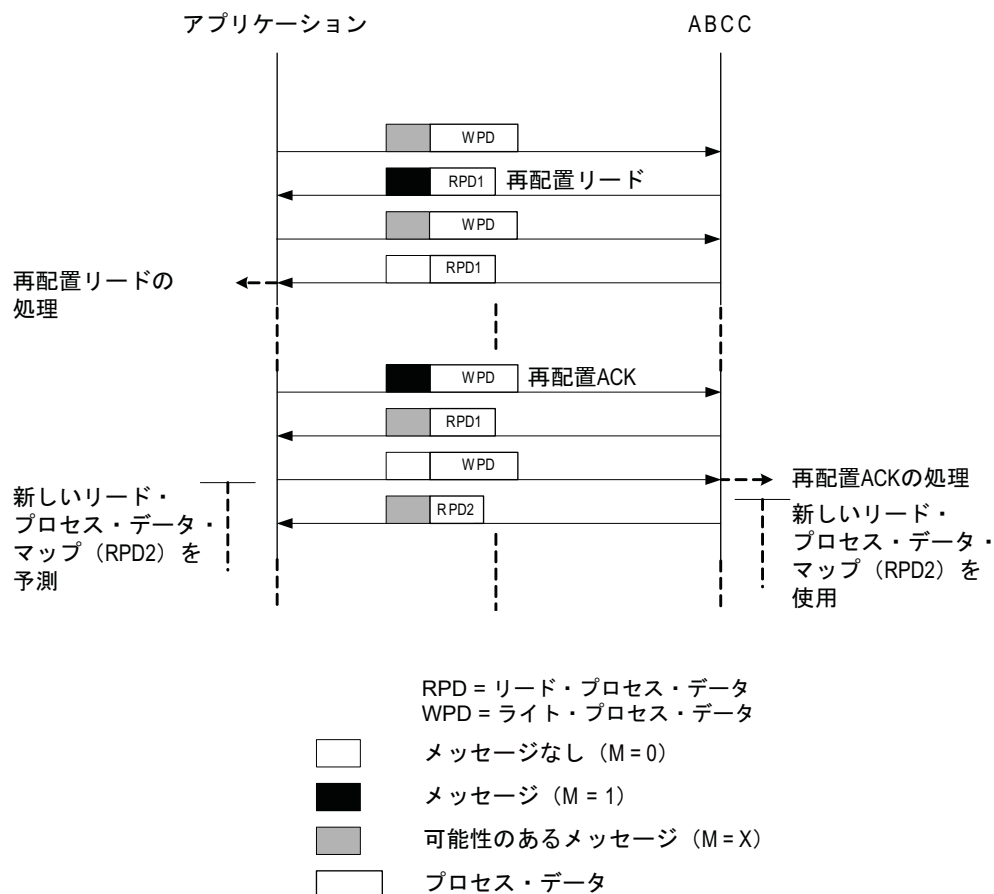
### D.1.2 ライト・プロセス・データ



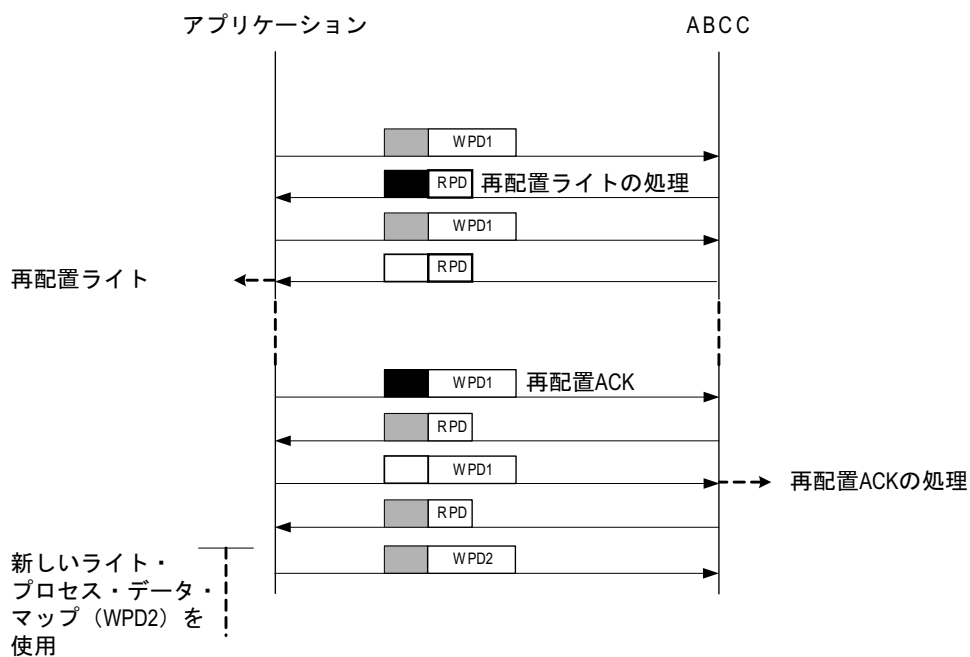
## D.2 シリアル・モード

テレグラムがピンポン方式で交換され、メッセージのないテレグラムが各コマンドを終了させることに注意してください。このため、いくつかのテレグラムは、再配置を行う前に交換する必要があります。

### D.2.1 リード・プロセス・データ



## D.2.2 ライト・プロセス・データ



RPD = リード・プロセス・データ  
 WPD = ライト・プロセス・データ

- メッセージなし (M=0)
- メッセージ (M=1)
- 可能性のあるメッセージ (M=X)
- プロセス・データ

## D.3 例 : Remap\_ADI\_Write\_Area

ADI

2	a	(1 * UINT8)
3	b	(1 * UINT16)
4	c   d   e	(3 * UINT16)
5	f   g   h   i	(4 * UINT8)
8	j   k   l   m	(4 * UINT8)
12	n	(1 * UINT8)

初期マッピング :

マッピング項目	0	1	2			3			
ADIの要素	a	b	c	d	e	f	g	h	i

コマンドRemap\_ADI\_Write\_Area :

CmdExt[0]	1	マッピング項目1から再配置を開始 (予約)
CmdExt[1]	0	
Data[0...1]	2	2つのマッピング項目 (1と2) を削除
Data[2...3]	2	2つのマッピング項目を挿入
Data[4...5]	8	新しいマッピング項目1 : インスタンス番号#8
Data[6]	1	新しいマッピング項目1 : エlement1 (k) からのマッピング
Data[7]	3	新しいマッピング項目1 : 3つのElement (k..m) のマッピング
Data[8...9]	12	新しいマッピング項目2 : インスタンス番号#12
Data[10]	0	新しいマッピング項目2 : Element0 (n) からのマッピング
Data[11]	1	新しいマッピング項目2 : 1つのElement (n) のマッピング

結果 :

マッピング項目	0	1			2	3			
ADIの要素	a	k	l	m	n	f	g	h	i

## E. CRC の計算

### E.1 概要

注意：次の情報は、シリアル・インターフェースを使用する場合にのみ適用されます。

受信側が送信エラーを検出できるようにするために、各シリアル・テレグラム・フレームには、16 ビットのサイクリック冗長検査が含まれています。

CRC は次のように計算されます。

1. 16 ビットレジスタに FFFFh をロードします (簡潔にするために、'CRC レジスタ' と呼ぶ)。
2. メッセージの最初のバイトと CRC レジスタのロー・オーダー・バイトとの XOR をとり、その結果を CRC レジスタに入れます。
3. CRC レジスタの 1 ビットを右に (LSB の方へ) シフトし、MSB をゼロ充てんします。
4. レジスタからシフト・アウトしたばかりの LSB を検査します。設定されている場合、CRC レジスタと多項式の値 A001h (1010 0000 0000 0001) との排他的論理和をとります。
5. 8 つのシフトが終了するまで、ステップ 3 および 4 を繰り返します。
6. メッセージからの次のバイトと CRC レジスタのロー・オーダー・バイトとの XOR をとり、その結果を CRC レジスタに入れます。
7. 完全なメッセージが処理されるまで、ステップ 3～6 を繰り返します。
8. CRC レジスタには、最終的な CRC16 の値が含まれています。



## E.2 例

この例では、速いアプローチを使用して CRC を計算します。全ての可能な CRC の値は、単にメッセージ・バッファを通した関数の増分としてインデックスされている 2 つの配列にあらかじめロードされます。

