

FAULHABER

通信マニュアル

MC 5010

MC 5005

MC 5004

MCS

EtherCAT[®]
Technology Group

The logo for EtherCAT Technology Group, featuring the word "EtherCAT" in a bold, black, sans-serif font with a registered trademark symbol. Above the "T" is a stylized grey arrow pointing right. Below "EtherCAT" is the text "Technology Group" in a smaller, black, sans-serif font.

JP

WE CREATE MOTION

インプリント

バージョン:

第2版、9-12-2016

Copyright

Dr. Fritz Faulhaber GmbH & Co. KG

Daimlerstr. 23 / 25 · 71101 Schönaich

翻訳を含む転載禁止。

本マニュアルの全部または一部を、Dr. Fritz Faulhaber GmbH & Co. KGの明示的な書面による同意なしに、情報システムへ複製、再生、保存、あるいは他の形式に加工または転送することは禁止されています。

本マニュアルは細心の注意を払って作成されています。

ただし、Dr. Fritz Faulhaber GmbH & Co. KGは本マニュアルの誤記および誤記によって発生した結果に対して何ら責任を負わないものとします。同様に、機器の不適切な使用による直接的損失および結果的損害に関しても何ら責任を負わないものとします。

本ソフトウェアを使用する際には、安全工学および干渉抑制に関連する規制ならびに本マニュアルに記載された要件に留意し遵守する必要があります。

仕様は予告なしに変更されることがあります。

最新版のテクニカルマニュアルは、以下のFAULHABERのインターネットサイトから入手できます：
www.faulhaber.com

目次

1	本マニュアルについて	5
1.1	本マニュアルの目的	5
1.2	関連文書	5
1.3	本書の使用	5
1.4	略語一覧	6
1.5	本マニュアルで使用する記号とマーク	7
2	概要	8
2.1	EtherCAT機器の基本構成	8
2.2	FAULHABER Motion Manager	9
2.3	通信の前提条件（物理層）	10
2.4	ESIファイル	11
2.5	スレーブの識別	11
3	EtherCAT通信	12
3.1	はじめに	12
3.2	データリンク層	12
3.2.1	EtherCATフレームおよびデータグラム	13
3.2.2	SyncManagerの管理	14
3.2.3	アドレス指定	15
3.2.4	アプリケーション層とのインターフェース	15
3.3	アプリケーション層	16
3.4	PDO（プロセスデータオブジェクト）	17
3.4.1	PDOの構成	17
3.4.2	標準構成のPDOマッピング（初期設定のステータス）	17
3.5	SDO（サービスデータオブジェクト）	18
3.5.1	SDOエラーの詳細	19
3.6	緊急オブジェクト（エラーメッセージ）	20
3.7	同期	23
3.7.1	分散クロックによる同期（DC-Sync）	23
3.7.2	SyncManagerイベントによる同期（SM-Sync）	24
3.8	層の管理	25
3.8.1	EtherCAT機器状態の制御	25
3.8.2	スレーブ情報インターフェース（SII）	26
3.9	オブジェクトディクショナリのエントリ	26
3.10	エラーハンドリング	27
3.10.1	装置不良	27
3.10.2	通信エラー	28
3.10.2.1	EtherCATフレームのエントリのチェックによるエラー検出	28
3.10.2.2	エラー応答	29
3.10.2.3	ネットワークトラフィック分析	30
3.10.2.4	EtherCAT AL Status Codeとトラブルシューティング	30

3.11	パラメータの保存および復元	32
3.11.1	パラメータの保存	32
3.11.2	設定の復元	33
3.11.3	パラメータセットの変更	33
4	追跡レコーダ	36
4.1	追跡設定	36
4.2	追跡バッファの読み込み	38
4.3	追跡機能の一般的な実行	39
5	パラメータの説明	40
5.1	CiA 301の通信オブジェクト	40
5.2	メーカー固有のオブジェクト	48

1 本マニュアルについて

1.1 本マニュアルの目的

本マニュアルでは次の内容を記載しています。

- 駆動機器とのEtherCAT通信
- 通信構築により提供される基本サービス
- パラメータのアクセス方法
- 通信システムから見た駆動機器

本マニュアルの読者は、EtherCATの知識があるソフトウェア開発者およびEtherCATプロジェクトのエンジニアが対象です。

本マニュアルの全ての情報は、標準バージョンの駆動機器を基本に説明しています。顧客固有のバージョンについては添付書類を参照してください。

また、本マニュアルの基本となるファームウェアのバージョンはGです。

1.2 関連文書

FAULHABER製品の設定および運転時の操作に関する追加情報は、以下のマニュアルを参照してください。

マニュアル	詳細
Motion Manager 6	FAULHABER Motion Manager PCソフトウェア取扱説明書
クイックスタートガイド	FAULHABERモーションコントローラの設定および運転の最初の手順の説明
機能マニュアル	駆動機器の運転モードおよび機能の説明
テクニカルマニュアル	FAULHABERモーションコントローラの設置および使用ガイド
CiA 301	CANopenアプリケーションレイヤおよび通信プロファイル
CiA 402	駆動機器およびモーションコントローラ用CANopen機器プロファイル

マニュアルは以下のサイトからpdf形式でダウンロードできます：www.faulhaber.com/manuals/

1.3 本書の使用

- ▶ 製品を構成する前に本マニュアルをよくお読みください。
- ▶ 本マニュアルは、製品の使用期間にわたって、オペレータがいつでも手に取って読むことができる場所に保管してください。
- ▶ また、製品を譲渡する際には製品と一緒に本マニュアルも次の所有者に渡してください。

1.4 略語一覧

略語	意味
AL	アプリケーション層
Attr.	属性
CAN	コントローラエリアネットワーク
CSP	サイクル同期位置モード
CSV	サイクル同期速度モード
DC	分散クロック
DL	データリンク層
EEPROM	電氣的消去可能プログラマブルROM
EMCY	緊急
ESC	EtherCATスレーブコントローラ
ESI	EtherCATスレーブ情報
ETG	EtherCATテクノロジーグループ
EtherCAT	制御オートメーション技術向けイーサネット
FCS	フレーム検査シーケンス
FMMU	フィールドバスメモリー管理ユニット
HB	上位バイト
HHB	高上位バイト
HLB	高下位バイト
LB	下位バイト
LHB	低上位バイト
LLB	低下位バイト
LSB	最下位バイト
LSS	レイヤー設定サービス
MSB	最上位バイト
OD	オブジェクトディクショナリ
PDO	プロセスデータオブジェクト
PP	プロファイル位置
PV	プロファイル速度
ro	読み取り専用
RTR	遠隔要求
rw	読み取り／書き込み
RxPDO	受信プロセスデータオブジェクト（駆動機器からPDOを受信）
SDO	サービスデータオブジェクト
SII	スレーブ情報インターフェース
PLC	プログラマブルロジックコントローラ - PLC
Sxx	符号付（正の数と負の数）データ型、ビットサイズxx
TxPDO	送信プロセスデータオブジェクト（駆動機器にPDOを送信）
Uxx	符号なし（正の数）データ型、ビットサイズxx

1.5 本マニュアルで使用する記号とマーク

注意!



機器が損傷する危険

- ▶ 上記の状況を回避するための措置



運転に対する理解を深めるヒントや適切な指示の情報

- ✓ 必要な操作の前提条件

1. 必要な操作の最初の手順

- ↳ 手順の結果

2. 必要な操作の次の手順

- ↳ 操作の結果

- ▶ 単発の操作

2 概要

EtherCATは、ドイツのBeckhoff Automation GmbHの登録商標および同社がライセンスを保有する特許技術です。

2.1 EtherCAT機器の基本構成

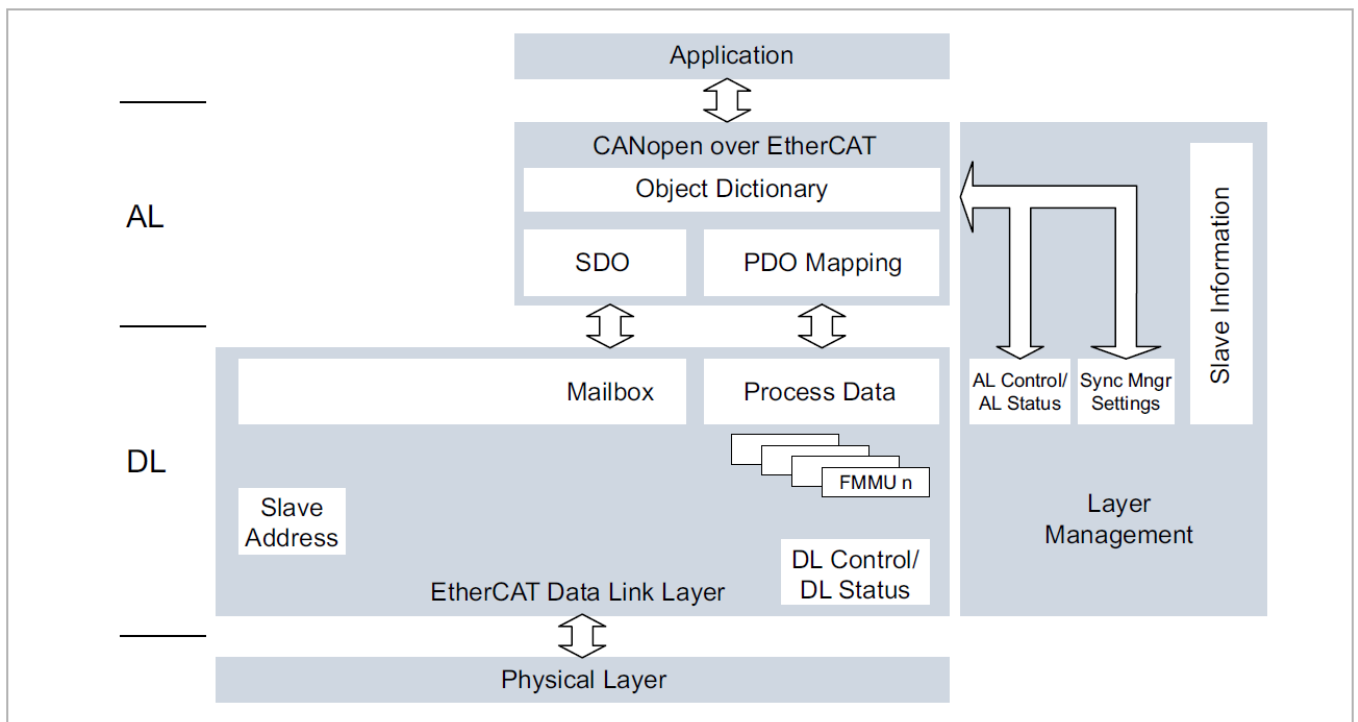


図1: EtherCAT機器の基本構成

物理層 (Physical Layer)

EtherCAT物理層の構成は、イーサネットの仕様であるIEEE 802.3に準拠しています。標準100Base-TXに対応するイーサネットで、EtherCATマスターとEtherCATスレーブをリンクします。物理層ではデータリンク層の更新するデータパケットにフレーム情報を追加して、データパケットを暗号化/復号化します。

データリンク層 (Data Link Layer)

EtherCATフレーム通過時に、データリンク層がデータを抽出/挿入するとともに、EtherCATフレームの完全性もチェックします。これらは、データリンク層のパラメータに保存された規則に従って実行されます。データは、EtherCATスレーブの各メモリーセクションで、メールボックスデータまたはプロセスデータとして利用可能になります (3.2節、12ページ参照)。

アプリケーション層 (Application Layer)

アプリケーション層には、データ層と駆動機器との通信に必要な全てのサービスとオブジェクトが含まれ、サービスはCANopenに基づき構成されます (3.2節、12ページ参照)。


アプリケーション (Application)

アプリケーション部にはドライブ機能が含まれ、CiA 402に対応しています。ドライブ機能は、オブジェクトディクショナリディクショナリから読み取ったパラメータにより、セットポイントを取得して実際の値を返します。オブジェクトディクショナリのパラメータは、駆動機器の動作を制御します。

i 本マニュアルでは、アプリケーション部の詳細な説明をしていません。駆動機器との通信および関連する動作モードについては、別冊「ドライブ機能」のマニュアルを参照してください。

2.2 FAULHABER Motion Manager

FAULHABER駆動機器の最初の設定は、モーションコントローラのUSBポートまたはシリアルCOMポート（どちらでも使用できます）を介し、「FAULHABER Motion Manager」ソフトウェアを使用して実行することをお勧めします。


-  複数のポートを同時に使用すると、許容できない過渡的な状態が発生する恐れがあります。
USBポートまたはRS232ポートを使用してFAULHABER駆動機器を構成する前に、EtherCATネットワークからモーションコントローラを切断してください。

FAULHABER Motion Managerを使用すると、接続されたモータコントローラの設定やパラメータに簡単にアクセスすることができます。グラフィカルユーザインターフェースからは、設定の読み込み、変更、リロードができます。また、個別のコマンド、パラメータセットおよびプログラムシーケンスをコントローラに入力しロードすることもできます。

駆動コントローラの設定を実行する場合は、ウィザード機能を使用できます。ウィザード機能はユーザインターフェースに配置され、常に同じ手順で実行できます。

- 接続ウィザード: 接続されたコントローラとの接続を確立します。
- モータウィザード: 各FAULHABERモータを選択することで、接続されたモータへの外部コントローラを設定します。
- コントロール設定ウィザード: コントロールパラメータを最適化します。

FAULHABERインターネットページから無料でソフトウェアをダウンロードできます。

-  常に最新のFAULHABER Motion Managerを使用することをお勧めします。

FAULHABER Motion Managerについては、別冊の「Motion Manager 6」マニュアルを参照してください。マニュアルの内容は、FAULHABER Motion Managerのオンラインヘルプからも参照できます。

2.3 通信の前提条件（物理層）

i ネットワーク接続ケーブルとして、カテゴリ5e（EN 50288によるCat5e）以上の最大長100mのイーサネットパッチケーブルまたはクロスオーバーケーブルが使用できます。

同じ物理ネットワーク内でEtherCATと標準イーサネットを絶対に一緒に使用しないでください。通信障害が発生する可能性があります。

i 複数のポートを同時に使用すると、許容できない過渡的な状態が発生する恐れがあります。

モーションコントローラをEtherCATネットワークに接続する前に、他のポート（USB、RS232など）に接続されていないことを確認してください。

1. コントローラを電源に接続します（最低限、回路電源への電源供給）。
2. EtherCATのINポートをマスター側ポートに接続します（図2参照）。
3. 複数のコントローラを使用している場合、各EtherCATのOUTポートを次のコントローラのEtherCATのINポートに接続します。
 - ↳ 最後のコントローラ（スレーブ）のEtherCATのOUTポートには何も接続しません。マスターから渡されたテレグラムが全てのスレーブを通過すると、テレグラムは同じケーブルを使用してマスターに戻ります。
4. 電源を投入します。
5. 設定アプリケーションを使用して接続を確立します（2.2節、9ページ参照）。
6. EtherCATスレーブ情報を定義します（2.4節、11ページ参照）。

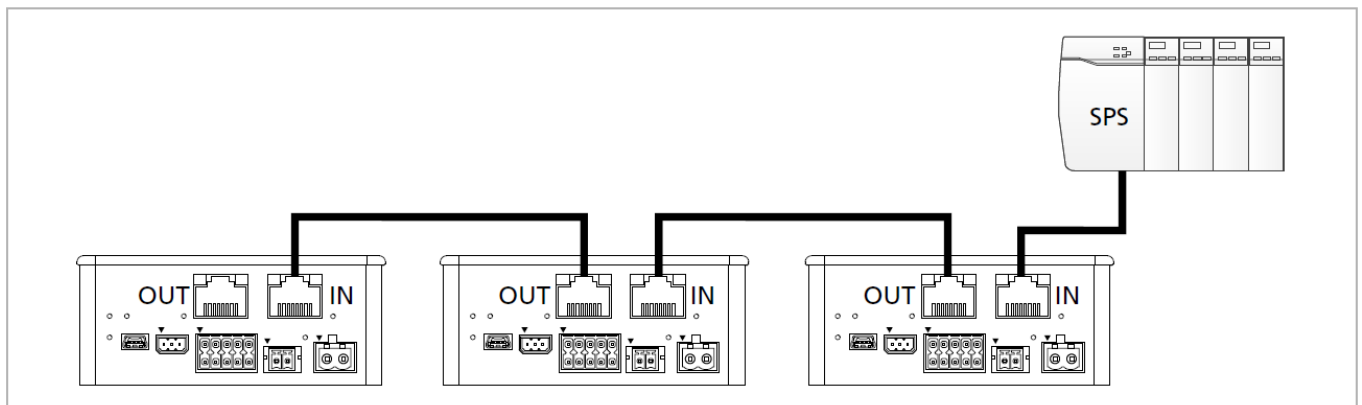


図2: EtherCATネットワーク接続

電源を投入し初期化されると、モーションコントローラは初期化状態になります。

ドライブ機能を実行可能にするには、モーションコントローラを運転状態にする必要があります。

2.4 ESIファイル

接続された駆動機器に関する情報とEtherCATスレーブとの通信に必要な動作は、ESI（EtherCATスレーブ情報）ファイルに記述されています。（必要な設定、データ型、オブジェクトディクショナリが含まれます。）

FAULHABERモーションコントローラのESIファイルは以下の場所にあります。

- XMLファイル形式の機器全体の情報はモーションコントローラに格納されています。ESIファイルは、直接FAULHABERから入手することもできます。
- モーションコントローラのEtherCAT EEPROMには、やや簡単に記述されたファイルが保存されています（3.8.2項「スレーブ情報インターフェース」、26ページ参照）。

適切に構成されたEtherCATマスターは、ESIファイルの情報を読み込むことができ、利用可能なESIファイルとネットワークの検出された駆動機器を比較します。メーカー番号（0147）、製品コード、改訂番号（該当する場合）が一致すると、駆動機器のESIファイルが特定され、マスターはESIファイルに保存された設定で駆動機器を構成できます。ESIファイルには複数の改訂番号が入力できるため、複数のファームウェアのバージョンに対応します。

サイクル時間

ESIファイルには、「Adapt Automatically」属性の「Cycle Time」のエントリが含まれています。EtherCATマスターは、ETG2000に従って動作するように、ここにサイクル時間を入力します。

この設定に関わらず、EtherCATマスターによるサイクル時間の自動的な入力に失敗した場合は、手動でエントリを修正する必要があります。これを行うには、初期値0をマスターのサイクル時間（ナノ秒）に置き換える必要があります。サイクル時間を4msに設定するには、値**4000000**と入力してください。

2.5 スレーブの識別

ネットワーク上でEtherCATマスターは以下のようにスレーブを識別します。

- 位置番号による識別:
各スレーブには、イーサネットセグメントのロジックな位置に従って番号が付けられ、その番号により識別できます。番号は、EtherCATマスターにより昇順で付けられ、EtherCATマスターの直後のスレーブが1、次のスレーブが2となります。
- 明示的な機器IDによる識別:
設定段階でオブジェクト2400.08（明示的な機器ID）の内容を任意の値に設定し、アプリケーションEEPROMに「SAVE」コマンドで保存します（3.11節、32ページ参照）。動作時にEtherCATマスターは、EtherCATメカニズムを使用してIDを読み込み、以前に保存された情報と比較します。これにより、交換された機器や切り替えられたケーブルが確実に検出されます。



FAULHABER Motion Managerを使用すると、明示的な機器IDの入力や保存が簡単にできます。

3 EtherCAT通信

3.1 はじめに

EtherCAT

EtherCATはイーサネットに基づいた通信技術です。EtherCATを使用するには、ネットワークおよび接続されたスレーブとの通信を制御するEtherCATマスターが必要です。EtherCATネットワークでは65,000台以上の機器をアドレス指定することができます。また、EtherCATでは全二重通信が使用され、最大100MBit/sの転送速度が実現します。

EtherCAT仕様

FAULHABER駆動機器にとって重要なETG仕様は、以下を定義しています。

- ETG1000シリーズ: EtherCATテクノロジーおよび通信構造
- ETG2000シリーズ: EtherCATスレーブ情報 (ESI) の仕様
- ETG6010: CiA 402駆動機器プロファイルの実装

以下のような幅広い機器クラスで、CANopen機器プロファイルが定義されています。

- 駆動機器用CiA 402
- 入出力機器用CiA 401

3.2 データリンク層

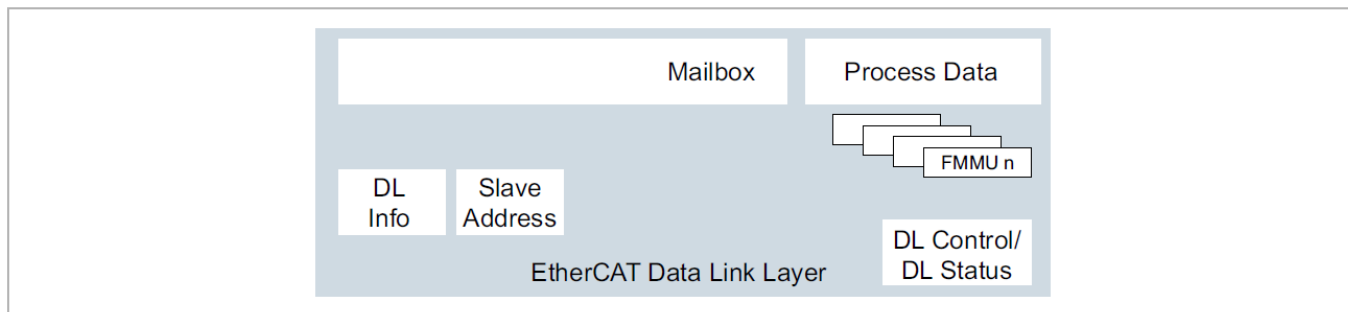


図3: データリンク層

データリンク層は、物理層とアプリケーション層を接続します。データリンク層の機能には、基本的に以下の制御および通信サービスが含まれています。

- 物理層とのインターフェース管理 (3.2.1項、13ページ参照)
- アプリケーション層とのインターフェース管理 (3.2.4項、15ページ参照)
- EtherCAT EEPROMへのアクセス
- ESC構成
- 分散クロック (3.7.1項、23ページ参照)
- EtherCATスレーブのアドレス指定 (3.2.3項、15ページ参照)
- SyncManagerの管理 (3.2.2項、14ページ参照)

3.2.1 EtherCATフレームおよびデータグラム

フレーム構造

EtherCATフレームは、最大1518バイトで構成され、最大1450バイトのユーザデータを格納できます。

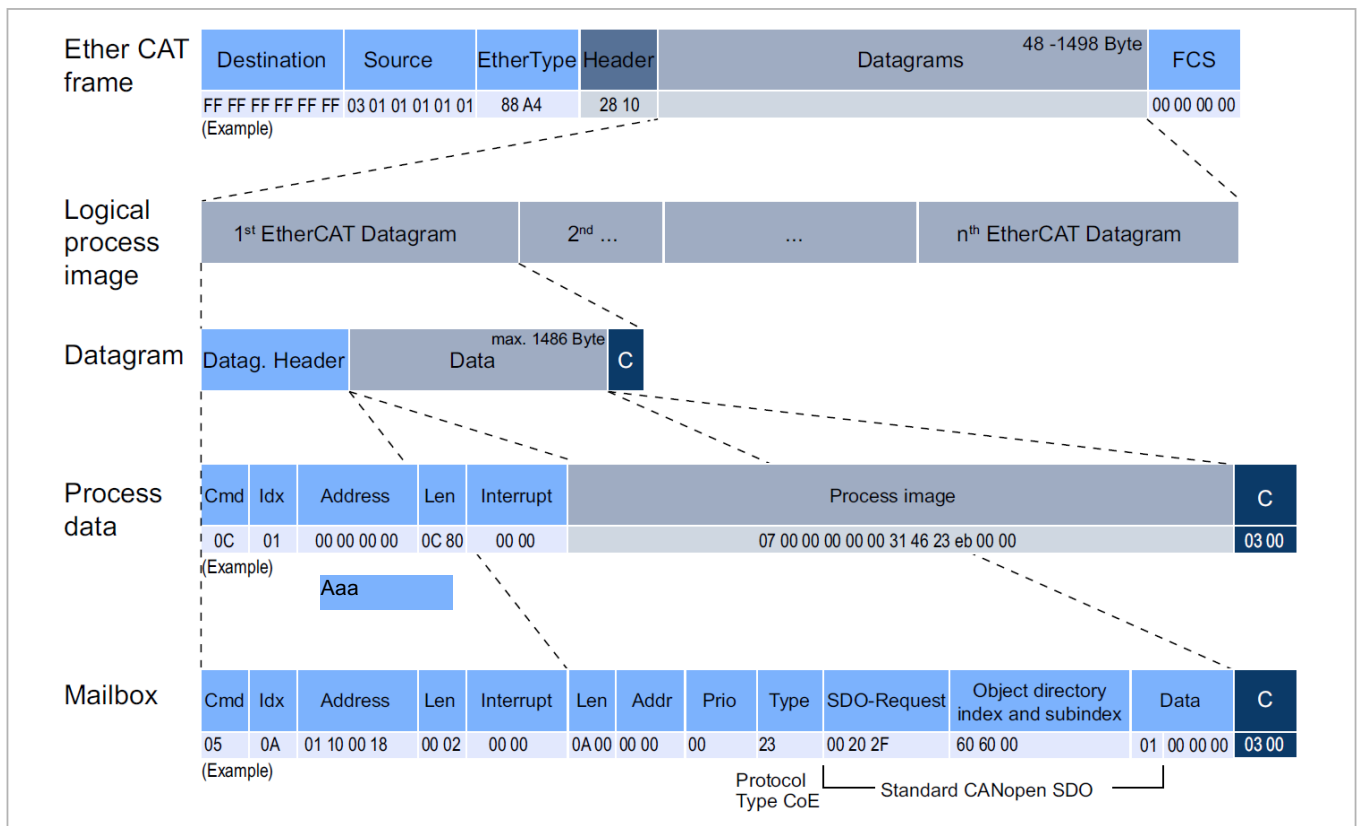


図4: フレーム構造

EtherCATフレームは以下のデータエリアで構成されています。

- **イーサネットヘッダー:** イーサネットヘッダーにはフレームの送信元アドレスおよび宛先アドレス、使用するプロトコルの種類が含まれます。
- **データグラム:** 1個以上のデータグラム（以下のデータグラムの説明を参照）。
- **フレーム検査シーケンス:** これらのデータは、エラーチェックに使用されます（3.10.2項、28ページ参照）。

データグラムは以下のデータエリアで構成されています。

- **データグラムヘッダー:** データグラムヘッダーには、通信の種類、メモリアクセス権、ユーザデータのアドレスと長さに関する情報が含まれます。
- **ユーザデータ:** ユーザデータの構造はメールボックスとプロセスデータ通信で異なります。CANopenで使用するサービスデータオブジェクト（SDO）またはプロセスデータオブジェクト（PDO）が含まれています。
- **ワーキングカウンタ:** ワーキングカウンタはデータ交換エラーを検出するために使用されます。（3.10.2項、28ページ参照）。

EtherCATスレーブ1台当たりのプロセスデータサイズにほぼ制限はなく、必要に応じて複数のデータグラムに分割できます。プロセスデータの構成はサイクルごとに異なる場合があります。

EtherCATフレームのパス

EtherCATマスターは、ペアケーブルの一番目（RX側）を使用して、EtherCATフレームを送信します。各フレームはネットワークの全てのEtherCATスレーブを通過します。EtherCATフレーム通過時に、EtherCATスレーブコントローラ（ESC）はデータを抽出／挿入できます。各ESCは常にEtherCATフレームの特定の部分についてのみ、常に読み込み／書き込みが可能です。ネットワークの最後のEtherCATスレーブは、ペアケーブルの二番目（TX側）を使用してEtherCATフレームをEtherCATマスターに戻します。

データシーケンスは、ネットワークのEtherCATスレーブの物理的シーケンスとは関係がありません。EtherCATスレーブ間のブロードキャスト、マルチキャストおよび相互通信も可能です。

3.2.2 SyncManagerの管理

SyncManagerによるデータ送信

EtherCATフレームからPDOおよびSDOを読み込む（パラメータ受信）、またはEtherCATフレームにPDOおよびSDOを組み込む（パラメータ送信）には、SyncManagerを使用します。データ送信には4個の同期チャンネルを使用できます。

SyncManagerチャンネル	機能
0	EtherCATフレームからメールボックスへのサービスデータの送信（SDO受信）
1	メールボックスからEtherCATフレームへのサービスデータの送信（SDO送信）
2	EtherCATフレームからプロセスデータの送信（PDO 1/2/3/4受信）
3	EtherCATフレームへのプロセスデータの送信（PDO 1/2/3/4送信）

プロセスデータの送信には、SyncManagerオブジェクト0x1C12および0x1C13が使用できます（5.1節、40ページ参照）。

読み込み／書き込みアクセスの監視

SyncManagerは、メモリーエリアの読み込み中に一部のメモリーエリアが上書きされないように、EtherCATマスターとEtherCATスレーブからのデータ交換メモリーへの同時アクセスを防止し、読み込むデータの整合を保持します。

データ交換には2種類のメモリーを使用できます。

メモリーの種類	詳細
Mailbox memory	<p>メールボックスメモリーには単一のメモリーエリアで構成されています。</p> <p>SyncManagerは以下の機能を実行します。</p> <ul style="list-style-type: none"> メモリーの書き込み中にメモリーの読み込みを防止。 メモリーの読み込み中にメモリーの書き込みを防止。 メモリーをオーバーフローから保護。 <p>このメモリーは、リアルタイムなデータに適していません。サービスデータのみを使用します。</p>
Buffer memory for process data	<p>バッファメモリーは3つのバッファエリアに分割されています。</p> <p>SyncManagerは以下の機能を実行します。</p> <ul style="list-style-type: none"> 現在書き込み中ではないバッファエリアが、書き込み用に選択されます。書き込み中はメモリーへの読み込みアクセスをブロック。 バッファエリアへの書き込み完了後、読み込みが可能。読み込み中はメモリーへの書き込みアクセスをブロック。 <p>以下のバッファエリアが常に利用できます。</p> <ul style="list-style-type: none"> バッファエリア1、現在書き込み中 バッファエリア2、現在読み込み中 バッファエリア3、書き込みが完了し読み込み可能（バッファエリア1の書き込みアクセスが完了する前にバッファエリア2の読み込みアクセスが完了した場合） <p>バッファエリア1の書き込みアクセスが完了する前にバッファエリア2の読み込みアクセスが完了した場合、バッファエリア1のデータは読み込み可能となり、バッファエリア3のデータは破棄されます。</p>

3.2.3 アドレス指定

EtherCATプロトコルを使用すると以下の手順でアドレス指定できます。

- 位置のアドレス指定: ネットワークのEtherCATスレーブの物理的な位置がアドレスの役割を果たします。EtherCATスレーブでは特定のメモリーエリアがアドレス用に確保されています。
- ノードのアドレス指定: 設定時に、EtherCATマスターによりEtherCATスレーブに割り当てられたノードアドレスが、アドレスとして機能します。EtherCATスレーブの特定のメモリーエリアがアドレス用に確保されています。
- 論理アドレス指定: ネットワークのメモリーエリア全体、すなわちEtherCATマスターと全てのEtherCATスレーブのメモリーエリアが、パラメータを使用してアドレス指定できる論理メモリー内に再現されます。EtherCATスレーブの物理アドレスの論理アドレスへの割り当ては、EtherCATマスターに保存されます。論理アドレスは、開始フェーズ中にデータリンク層のフィールドバスメモリー管理ユニット (FMMU) に転送され、物理アドレスに変換されます。

3.2.4 アプリケーション層とのインターフェース

表1: データリンクとアプリケーション層とのインターフェース

インターフェース	詳細
Mailbox	<p>メールボックスは、時間が優先課題ではないデータ専用で使用されます。時間が優先課題ではないデータにはサービスデータがあります。</p> <p>サービスデータは、CANopen (CiA 301) に基づきサービスデータオブジェクトフレーム (SDOフレーム) を使用して送信されます (3.5節、18ページ参照)。サービスデータの送信は順次実行されます。</p>
Process data	<p>プロセスデータはリアルタイムなデータです。それは、常に最新のデータにアクセスできることを意味します。呼び出されないデータ (サイクル時間などEtherCATスレーブでは素早く処理できないデータ) は破棄されます。</p> <p>プロセスデータは、CANopen (CiA 301) に基づきプロセスデータオブジェクトフレーム (PDOフレーム) を使用して送信されます (3.4節、17ページ参照)。サービスデータの送信は周期的に実行されます。</p>

3.3 アプリケーション層

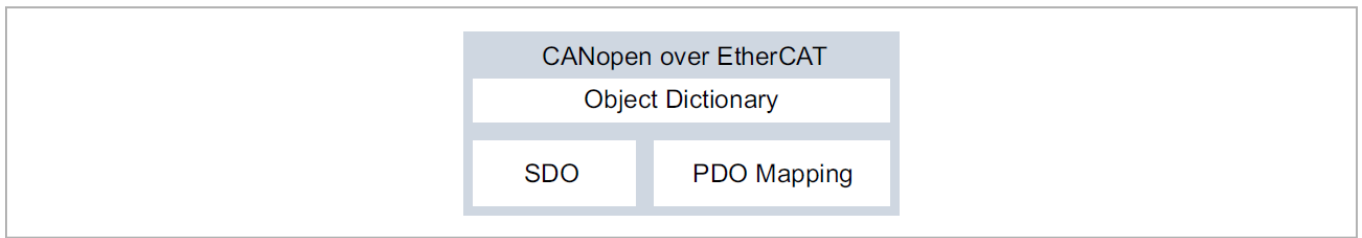


図5: アプリケーション層

EtherCAT上のCANopen

FAULHABERモーションコントローラは、CiA 301に準拠したCANopen通信プロファイルによりEtherCAT (CoE) プロトコルのCANopen機器をサポートしています。

- 4個の送信PDO (TxPDO)
- 4個の受信PDO (RxPDO)
- 2個のSDO

基本となるCANテレグラムが8バイトであるのに対し、CANopenテレグラムは250バイト長まで使用できるため、より多くの記述ができます。

CiA 402に準拠したCANopen駆動機器プロファイルは、そのままEtherCATで使用できます (機能マニュアル参照)。

オブジェクトディクショナリ

オブジェクトディクショナリには、駆動機器の設定値と実際の値が含まれ、アプリケーション (ドライブ機能) と通信サービス間をリンクします。オブジェクトディクショナリ内の全てのオブジェクトは、16ビットのインデックス番号 (0x1000~0x6FFF) と8ビットのサブインデックス (0x00~0xFF) で、アドレス指定できます。

インデックス	オブジェクトの割り当て
0x1000 to 0x1FFF	通信オブジェクト
0x2000 to 0x5FFF	メーカー固有のオブジェクト
0x6000 to 0x6FFF	CiA 402駆動機器プロファイルのオブジェクト

パラメータの値は、通信側または駆動機器側で変更できます。

通信部にはCiA 301で規定する通信サービスが含まれています。

PDOのデータ割り当ては、CiA 402で規定する「PDO set for servo drive (サーボドライブ用PDOセット)」にあらかじめ設定されています。

3.4 PDO（プロセスデータオブジェクト）

PDOには、駆動機器の動作を制御および監視するプロセスデータが含まれています。駆動機器は受信PDOと送信PDOを区別します。

- 受信PDO（RxPDO）：駆動機器で受信され、一般的に制御データが含まれています。
- 送信PDO（TxPDO）：駆動機器により送信され、一般的に監視データが含まれています。
PDOは駆動機器がNMT 運転状態の場合のみ評価または送信されます（3.8.1項、25ページ参照）。

3.4.1 PDOの構成

- 1つのPDOに対して最大4個のパラメータをマッピングできます。
- PDOのデータ配置は、オブジェクト0x1600～0x1603および0x1A00～0x1A03を使用して変更することができます。必要なマッピング手順は、CiA 301を参照してください。マッピング手順の実行には、適切なツール（FAULHABER Motion ManagerまたはPLCコントローラのシステムマネージャなどを使用していない場合）が必要です。

3.4.2 標準構成のPDOマッピング（初期設定のステータス）

RxPDO1: 制御ワード

2バイトのユーザーデータ					
LB	HB				

RxPDO1には、CiA DSP402に準拠した16ビットの制御ワードが含まれています。制御ワードは、駆動ユニットの機器状態を制御し、オブジェクトディクショナリのオブジェクトインデックス0x6040を指定します。ビット配分については、ドライブ機能のマニュアルを参照してください。

TxPDO1: 状態ワード

2バイトのユーザーデータ					
LB	HB				

TxPDO1には、CiA 402に準拠した16ビットの状態ワードが含まれています。状態ワードは、駆動ユニットの状態を示し、オブジェクトディクショナリのオブジェクトインデックス0x6041を指定します。ビット配分については、ドライブ機能のマニュアルを参照してください。

RxPDO2: 制御ワード、目標位置（PPおよびCSP）

6バイトのユーザーデータ					
LB	HB	LLB	LHB	HLB	HHB

RxPDO2には、16ビットの制御ワードおよびプロファイル位置モード（PP）の指定位置（オブジェクト0x607A）の32ビットの値が含まれています。

TxPDO2: 状態ワード、実位置値

6バイトのユーザーデータ					
LB	HB	LLB	LHB	HLB	HHB

TxPDO2には、16ビットの状態ワードおよび実位置値（オブジェクト0x6064）の32ビットの値が含まれています。

RxPDO3: 制御ワード、目標速度 (PVおよびCSV)

6バイトのユーザーデータ					
LB	HB	LLB	LHB	HLB	HHB

RxPDO3には、16ビットの制御ワードおよびプロファイル速度モード (PV) の設定速度 (オブジェクト0x60FF) の32ビットの値が含まれています。

TxPDO3: 状態ワード、実速度値

6バイトのユーザーデータ					
LB	HB	LLB	LHB	HLB	HHB

TxPDO3には、16ビットの状態ワードおよび実速度 (オブジェクト0x606C) の32ビットの値が含まれています。

RxPDO4: 制御ワード、目標トルク (PVおよびCSV)

6バイトのユーザーデータ					
LB	HB	LLB	LHB	HLB	HHB

RxPDO4には、16ビットの制御ワードおよび周期的トルクモード (CST) の目標トルク (オブジェクト0x6071) の16ビットの値が含まれています。

TxPDO4: 状態ワード、実トルク値

6バイトのユーザーデータ					
LB	HB	LLB	LHB	HLB	HHB

RxPDO4には、16ビットの状態ワードおよび周期的トルクモード (CST) の実トルク (オブジェクト0x6077) の16ビットの値が含まれています。

3.5 SDO (サービスデータオブジェクト)

SDOは、OD (オブジェクトディクショナリ) のパラメータを読み込んで記述します。SDOは16ビットのインデックスおよび8ビットのサブインデックスで、オブジェクトディクショナリにアクセスします。マスター (PC、PLC) の要求に応じて、モーションコントローラはデータを利用可能にする (アップロード) か、マスターからのデータを受信します (ダウンロード)。

表2: SDOユーザーデータの全般的な構成

バイト0	バイト1~2	バイト3	バイト4~7
Command specifier	16-bit index	8-bit subindex	4-byte parameter data

表3: 送信のSDOタイプの分散

送信の種類	バイト数	目的
Expedited transfer	最大 250 bytes	-
Segmented transfer	任意のサイズ	データブロックの送信 (追跡バッファなど)

送信の種類については、CiA 301を参照してください。

3.5.1 SDOエラーの詳細

SDOプロトコルをこれ以上処理できない場合、SDO-Abortテレグラムが送信されます。エラーの種類は、以下のようコード化されます（表2参照）

- バイト4 + 5: 追加エラーコードLB + HB
- バイト6: エラーコード
- バイト7: エラークラス

エラークラス	エラーコード	追加コード	詳細
0x05	0x03	0x0000	トグルビットが変更されない
0x05	0x04	0x0001	SDOコマンド指定子が無効または不明
0x06	0x01	0x0000	対象のオブジェクトへのアクセスがサポートされていない
0x06	0x01	0x0001	書き込み専用パラメータの読み込みの試行
0x06	0x01	0x0002	読み込み専用パラメータの書き込みの試行
0x06	0x02	0x0000	オブジェクトディクショナリにオブジェクトがない
0x06	0x04	0x0041	PDOにマッピングできないオブジェクト
0x06	0x04	0x0042	マッピングされたオブジェクトの数および/または長さがPDOの長さを超えている。
0x06	0x04	0x0043	全般的なパラメータの非互換性
0x06	0x04	0x0047	駆動機器の全般的な内部非互換性エラー
0x06	0x07	0x0010	データの種類またはパラメータ長が不一致または不明
0x06	0x07	0x0012	データの種類の不一致、パラメータ長が長過ぎる
0x06	0x07	0x0013	データの種類の不一致、パラメータ長が短過ぎる
0x06	0x09	0x0011	サブインデックスが存在しない
0x06	0x09	0x0030	全般的な値範囲エラー
0x06	0x09	0x0031	全般的な範囲エラー:パラメータ値が大き過ぎる
0x06	0x09	0x0032	全般的な範囲エラー:パラメータ値が小さ過ぎる
0x06	0x09	0x0036	全般的な範囲エラー: 最大値が最小値より大きい
0x08	0x00	0x0000	全般的なSDOエラー
0x08	0x00	0x0020	アクセス不可
0x08	0x00	0x0022	現在の機器状態ではアクセス不可

3.6 緊急オブジェクト（エラーメッセージ）

緊急メッセージはCANopen下にあるため、スレーブから自発的に送信されません。代わりにEtherCATマスターがメールボックスプロトコルを使用して要求する必要があります。しかし、この方法は時間がかかるため、緊急時の使用はお勧めしません。より良い手順としてエラーレジスタ1001hまたはFaulhaberエラーレジスタ2320hをPDOにマッピングする方法をお勧めします。これにより、マスターは最短の時間でエラー情報を受け取ることができます。

緊急オブジェクトは常に8バイトのサイズです。

8バイトのユーザデータ							
Error0 (LB)	Error1 (HB)	Error register	FE0 (LB)	FE1 (HB)	0	0	0

ユーザデータの割り当て:

- Error0(LB)/Error1(HB): 16ビットエラーコード
- エラーレジスタ: エラーレジスタ（オブジェクト0x1001の内容、5.2節、48ページ参照）
- FE0(LB)/FE1(HB): 16ビットFAULHABERエラーレジスタ（オブジェクト0x2320の内容、表7参照）
- 5~7バイト: 未使用（0）

エラーレジスタはエラーの種類を識別します。各エラーの種類はビットコード化され、それぞれのエラーコードが割り当てられています。エラーレジスタの最後の値は、オブジェクト0x1001で問い合わせすることができます。

最大3個の緊急メッセージを保存できます。EtherCATマスターが緊急メッセージを要求しない場合は、最も古い3個の緊急メッセージが保存され、その後に登録されたメッセージは破棄されます。これにより後続のエラーの原因となったエラーを検出できます。

表4は、各エラーはFAULHABERエラーレジスタの緊急マスクに含まれていれば（表8参照）、緊急メッセージで報告されたエラーのリストを示します。。

表4: 緊急エラーコード

緊急メッセージ		FAULHABERエラーレジスタ0x2320			エラーレジスタ0x1001	
エラーコード	指定	エラーマスク 0x2321	ビット	指定	ビット	指定
0x0000	エラーなし（エラーが無くなった時、または確認済みの時に送信されます）	-	-	-	-	-
-	-	-	-	-	0	汎用エラー（エラービット1~7のいずれかがセットされている場合にセットされます。）
0x3210	過電圧	0x0004	2	過電圧エラー	2	電圧エラー
0x3220	低電圧	0x0008	3	低電圧エラー	2	電圧エラー
0x43F0	温度警告	0x0010	4	温度警告	1	電流エラー ^{a)}
0x4310	温度エラー	0x0020	5	温度エラー	3	温度エラー
0x5410	出カステージ	0x0080	7	IntHW エラー	7	製造元固有のエラー

緊急メッセージ		FAULHABERエラーレジスタ0x2320			エラーレジスタ0x1001	
エラーコード	指定	エラーマスク 0x2321	ビット	指定	ビット	指定
0x5530	EEPROM 不具合	0x0400	10	メモリーエラー	-	-

緊急メッセージ		FAULHABERエラーレジスタ0x2320			エラーレジスタ0x1001	
エラーコード	指定	エラーマスク0x2321	ビット	指定	ビット	指定
0x6100	ソフトウェアエラー	0x1000	12	計算エラー	7	製造元固有のエラー
0x7200	測定回路：電流測定	0x0200	9	電流測定エラー	7	製造元固有のエラー
0x7300	Sensor fault (encoder) センサー不具合（エンコーダ）	0x0040	6	エンコーダエラー	7	製造元固有のエラー
0x7400	計算回路:モジュールの不具合	0x0100	8	モジュールエラー	7	製造元固有のエラー
0x8110	CAN overrun	0x0800	11	通信エラー	4	通信エラー
0x8130	CAN guarding failed					
0x8140	CAN recovered from bus stop					
0x8310	RS232 overrun					
0x84F0	偏差エラー(速度コントローラ)	0x0001	0	速度偏差エラー	5	駆動機器固有のエラー
0x84FF	最大速度エラー	0x2000	13	ダイナミックエラー	7	製造元固有のエラー
0x8611	追従エラー	0x0002	1	追従エラー	5	駆動機器固有のエラー

- a) 電流レギュレータは、モータの電流を常時指定範囲以下に維持します。警告温度以上になると、過電流エラービットが設定されます。その後、モータの許容電流がピーク電流値から連続電流値に減少します。

例:

表5のユーザデータが割り当てられた緊急メッセージは、以下のイベントで送信されます。

- エラーマスク0x2321に、ビット1（以下のエラー）がサブインデックス1（緊急マスク）の下にセットされず（表9参照）。
- オブジェクト0x6065.00にセットされた位置レギュレータ範囲内の制御偏差値が、オブジェクト0x6066.00のエラー遅延時間に設定された値により定義された時間を超えました（ドライブ機能のマニュアルを参照してください）。

表5: 緊急メッセージのユーザデータ割り当て例

8バイトのユーザデータ							
0x11	0x86	0x20	0x02	0x00	0x00	0x00	0x00

3.7 同期

FAULHABERモーションコントローラは、分散クロックおよびSyncManagerイベントを使用して同期をサポートしています。同期の種類は、受信PDOの場合はオブジェクト0x1C32.01、送信PDOの場合はオブジェクト0x1C33.01を使用して選択されます。値は以下のようになります。

- 0: 同期なし (FreeRun) 、EtherCATスレーブはサイクル時間0x1C32.02によって設定される自身のクロックに従って独自に動作します。
- 1または34: SyncManagerイベントによる同期 (3.7.2項、24ページ参照) 。
- 2: 分散クロックによる同期 (3.7.1項、23ページ参照) 。

以下の組み合わせのみが可能です。

0x1C32.01 SM2		0x1C33.01		
プロセスデータの受信				
00 h	FreeRun	00 h	FreeRun	サイクル時間のチェックなし
01 h	SM synchronous	22 h	SM synchronised with SM2	値 > 0の場合、サイクル時間の監視あり
02 h	DC Sync0	02 h	DC Sync0	サイクル時間の監視あり

マスターにより生成されるサイクル時間は、常に500 μsの倍数である必要があります。『SM synchronous』モードおよび『FreeRun』モードの最小サイクル時間は1msが指定されています。DC同期モードの最小サイクル時間は500 μsです。

SyncManagerをシンプル化するために、ESIファイルには2個のロットが含まれております。、これは、モーションコントローラに『DC Synchronous』モードと『SMSynchronous』モードの2つの動作モードのオプションがあることを、マスターに通知されます。動作モードは一度に1つのみをアクティブにできます。マスターがロットの概念をサポートしている場合は、必要なロットを選択するだけで、適切なSM構成を簡易的にかつエラーなしで生成することができます。

3.7.1 分散クロックによる同期 (DC-Sync)

各EtherCATスレーブには、ESCで管理される独自のクロックがあり、一番目のEtherCATスレーブ (参照スレーブ) の時間がネットワーク全体の参照時間となります。その他の全てのEtherCATスレーブおよびEtherCATマスターのクロックは、この参照時間から導き出されます。

そのために、EtherCATマスターは、参照クロックのEtherCATスレーブが現在の時間を入力する特別なデータグラムを頻繁に送信します。その他の全てのEtherCATスレーブおよびEtherCATマスターは、送信されたデータグラムの時間を読み込み、クロックを同期させます。ネットワークのEtherCATに接続された機器は論理リング構造で配列されているため、EtherCATマスターの後ろの一番目のEtherCATスレーブが参照スレーブになります。

EtherCATに接続された各機器は常に、読み込んだ参照クロックからの参照時間にデータグラムの伝播された時間を加味して修正します。EtherCATマスターは、伝播された時間を特別なデータグラムとしてEtherCATスレーブに送信します。このデータグラムを受信したESCは、受信時間をデータグラムに書き込みます。EtherCATマスターはその受信時間を読み込み、それをもとに適切な計算を実行します。

駆動機器のESCには、参照スレーブのマスタークロックと同期した内部マスタークロックがあります。クロックの同期では、テレグラムの伝播された時間が許容されます。この内部マスタークロックは、駆動機器のローカルサイクルを開始するSync0信号を生成します。

ローカルサイクルでは以前に受信し一時的に保存されたEtherCATテレグラムのプロセスデータが必要です。Sync0信号でローカルサイクルを開始すると、ローカルサイクルは保存されたプロセスデータを読み込み、制御ループを実行します。最後に入力データをプロセスイメージに書き込み、マスターが利用できるようにします。

マスターは、スレーブのサイクル時間と同じサイクル内でテレグラムを送信する必要があるため、スレーブからの現在のデータが常に処理されるデータとなります。マスターのサイクルでジッターが発生してパケットの設定が遅れた場合、現在の制御サイクルでは処理できないため、保留状態にして次の制御サイクルで処理する必要があります。この場合、現在の制御サイクルは1つ前のテレグラムのデータを使用します。

DCサイクル時間はオブジェクト0x1C32.02によって設定されず、代わりにマスターがESCレジスタに直接設定します。DCサイクル時間は、500 μ s以上またはその倍数に設定する必要があります。

3.7.2 SyncManagerイベントによる同期 (SM-Sync)

EtherCATスレーブのローカルサイクルは、プロセスデータテレグラムの受信時 (SyncManagerイベント) に開始されます。送信PDOが周期的に送信されている場合、EtherCATスレーブはSyncManager2イベント (SM2イベント) 発生時に同期化されます。受信PDOのみが周期的に送信されている場合、EtherCATスレーブはSyncManager3イベント (SM3イベント) 発生時に同期化されます。

SyncManagerイベントによる同期のパラメータは、『*運転前*』状態でオブジェクト0x1C32 (SM2) および0x1C33 (SM3) によって設定されます (5.1節、40ページ参照)。

プロセスデータエントリの監視

0x1C32.02 (サイクル時間) のエントリの目的は、マスターが送信したテレグラムを監視することです。プロセスデータは指定した時間内にスレーブに到着する必要があります。不良 (断線など) が発生し、データがスレーブに到着しない場合、スレーブが適切に構成されていると、スレーブは緊急メッセージを出力し『エラー』状態に切り替わります。サイクル時間が0に設定されている場合は、監視機能が非アクティブになります。

3.8 層の管理

層の管理より以下のサービスが可能です。

- EtherCAT機器状態の制御（3.8.1項、25ページ参照）
- スレーブ情報インターフェースでの読み込み／書き込み（3.8.2項、26ページ参照）

これらの機能を実行するために、EtherCATマスターはESCと直接通信します。

3.8.1 EtherCAT機器状態の制御

電源を投入し初期化されると、モーションコントローラは**運転前状態**になります。**運転前状態**では、機器はメールボックス通信のみを使用して機器と通信できます。

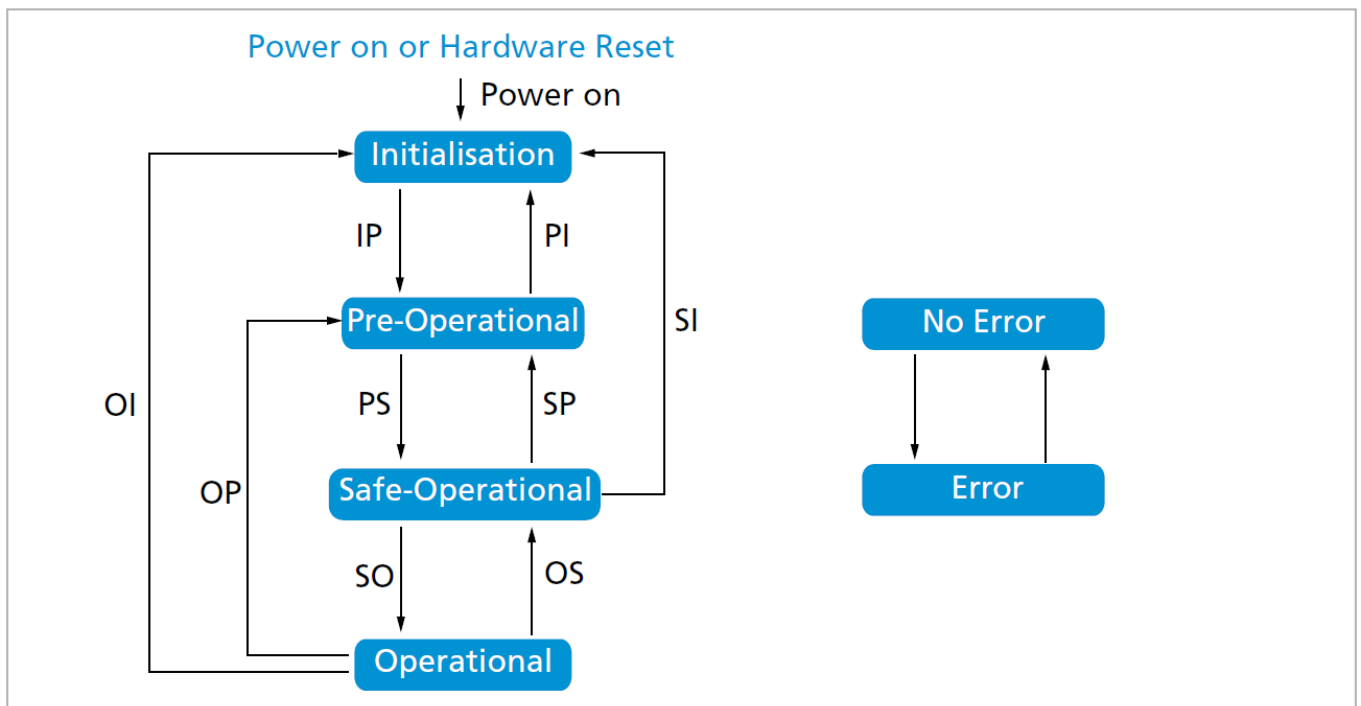


図6: EtherCAT機器状態

表6: 状態の変化

遷移	動作
Power on	<ul style="list-style-type: none"> 電源投入時に自動で初期化状態になります。 メールボックス通信、プロセスデータ通信はどちらも利用できません。 EtherCATマスターがメールボックス通信用のSyncManagerチャンネルを初期化します。
IP	<ul style="list-style-type: none"> EtherCATマスターがEtherCATフィールドバスを同期します。 EtherCATマスターがプロセスデータ通信用SyncManagerチャンネル、FMMUチャンネルおよびSyncManager-PDO割り当てを初期化します。 EtherCATマスターとEtherCATスレーブ間でメールボックス通信が確立します。 プロセスデータ送信の設定が送信されます。
PS	<ul style="list-style-type: none"> EtherCATスレーブが、プロセスデータ通信用SyncManagerチャンネルおよび分散クロック用設定が正しいかチェックします。 EtherCATスレーブがESCのメモリーエリアに現在の入力データをコピーします。 メールボックス通信およびプロセスデータ通信が利用可能になりました。EtherCATスレーブからの出力は安全状態となり、出力されません。入力データは周期的に更新されます。

遷移	動作
SO	<ul style="list-style-type: none"> ▪ EtherCATマスターが有効な出力データをEtherCATスレーブに送信します。 ▪ EtherCATマスターがEtherCATスレーブを運転状態に切り替えます。 ▪ 運転状態になるとEtherCATスレーブは入力データを出力にコピーします。 ▪ メールボックス通信およびプロセスデータ通信が利用可能になりました。

FAULHABERモーションコントローラのESIファイルには、全てのオブジェクトの初期設定が含まれています（2.4節、11ページ参照）。システム開始時のほとんどの場合、これ以上のパラメータ設定は必要ありません。

必要なパラメータ設定は、FAULHABER Motion ManagerによってUSBポートを使用して実行し、恒久的にEEPROMに保存できます（3.11節、32ページ参照）。EEPROMの設定はシステム開始直後から利用できます。

i 初期化状態では、駆動機器の全ての値は電源投入時の値にリセットされます。「Save」コマンド1010hで保存していないユーザの設定値は上書きされます。上書きしないようにするには、駆動機器を**初期化状態**に切り替えず、**運転前状態**にしてください。

i 駆動機器は駆動機器プロファイルのオブジェクト（制御ワード、状態ワード）で制御されます。駆動機器との通信および関連する動作モードについては、「ドライブ機能」のマニュアルを参照してください。

運転前状態への切り替えは、わずか数ミリ秒で実行されます。マスターはALレジスタ（130h）に問い合わせ、状態が正常に切り替わるまで待つ必要があります。正常に切り替わるまで、SDOは送受信されません。

3.8.2 スレーブ情報インターフェース（SII）

スレーブ情報インターフェースには、EtherCATスレーブおよび接続された駆動機器（オブジェクト0x1018の値など）の構成の固有のデータ、さらにメールボックスSyncManagerの固有のデータが含まれます。

これらのデータはEtherCAT EEPROMに保存され、ネットワークの設定で読み込まれます（2.4節、11ページ参照）。

3.9 オブジェクトディクショナリのエントリ

オブジェクトディクショナリは3つのエリアに分けられ、設定パラメータを管理します。各オブジェクトはインデックスとサブインデックス(SDOプロトコル)により参照できます。

- 通信パラメータエリア（インデックス0x1000～0x1FFF）には、CiA 301の通信オブジェクトが含まれます（5.1節、40ページ参照）。
- メーカー固有のエリア（インデックス0x2000～0x5FFF）には、メーカー固有のオブジェクトが含まれます（5.2節、48ページ参照）。
- 標準化機器プロファイルエリア（0x6000～0x9FFF）には、モーションコントローラによってサポートされたオブジェクトが含まれます（ドライブ機能のマニュアルを参照してください）。

3.10 エラーハンドリング

3.10.1 装置不良

表7: FAULHABERエラーレジスタ (0x2320)

インデックス	サブインデックス	名前	型	属性	デフォルト値	意味
0x2320	0x00	Fault register	U16	ro	-	FAULHABERエラーレジスタ

FAULHABERエラーレジスタは、ビットコード化された最新のエラーを含みます。エラーマスクオブジェクト (0x2321) で対象のエラー種類を選択し、エラーをマスクできます。

表8: エラーコード

エラービット	エラーメッセージ	詳細
0x0001	Speed deviation error	速度偏差が大きすぎる
0x0002	Following error	追従エラー
0x0004	Overvoltage error	過電圧検出
0x0008	Undervoltage error	不足電圧検出
0x0010	Temperature warning	警告出力時の温度を超えた
0x0020	Temperature error	エラーメッセージ出力時の温度を超えた
0x0040	Encoder error	エンコーダーのエラー検出
0x0080	IntHW error	内部ハードウェアのエラー
0x0100	Module error	外部モジュールのエラー
0x0200	Current measurement error	電流測定エラー
0x0400	Memory error	メモリーエラー (EEPROM)
0x0800	Communications error	通信エラー
0x1000	Calculation error	内部ソフトウェアエラー
0x2000	DynamicError	現在のモータ速度が設定最大速度より速い
0x4000	-	未使用、値 = 0
0x8000	-	未使用、値 = 0

上記のエラーは全て緊急エラーコードに対応しています (3.6節、20ページ参照)。

エラーマスクは、エラーコードにより内部エラーのハンドリングを記述します (表8参照)。

表9: エラーマスク (0x2321)

インデックス	サブインデックス	名前	型	属性	デフォルト値	意味
0x2321	0x00	Number of entries	U8	ro	6	オブジェクトエントリの数
	0x01	Emergency mask	U16	rw	0x00FF	エラーメッセージが送信されたエラー
	0x02	Fault mask	U16	rw	0x0000	駆動機器のを <i>Fault Reaction Active</i> 状態に切り替えるエラー
	0x03	Error Out mask	U16	rw	0x00FF	エラー出力ピンがセットされたエラー
	0x04	Disable voltage mask	U16	ro	0x0000	駆動機器をオフにするエラー (設定不可)
	0x05	Disable voltage user mask	U16	rw	0x0000	駆動機器をオフにするエラー (設定可)
	0x06	Quick stop mask	U16	rw	0x0000	駆動機器を <i>Quick Stop Active</i> 状態に切り替えるエラー

例:

- オブジェクト0x2321の不良マスク (サブインデックス2) が0x0001にセットされると、エラー状態は過電流に設定されるため、駆動機器はオフになります。
- オブジェクト0x2321のサブインデックス3が0にセットされると、エラー出力 (不良ピン) はエラーなしを示します。オブジェクト0x2321のサブインデックス3が0xFFFFにセットされると、エラー出力 (不良ピン) は全てのエラーを示します。

3.10.2 通信エラー

ネットワークの通信データエラーおよびデータ喪失を監視します。エラーが発生した場合、この手順を使用して接続された機器を安全な状態にしてエラーメッセージを出力することができます。ネットワークトラフィック分析を実行し、エラーを特定し修正する必要があります。

3.10.2.1 EtherCATフレームのエントリのチェックによるエラー検出

EtherCATスレーブとEtherCATマスターは直接通信できないため、EtherCATフレームのエントリを使用して不良データの監視をします。まず。

- フレーム検査シーケンス (FCS) : ESCはチェックサムを使用してEtherCATフレームの通過時に検査します。検査結果が正の場合にのみ、EtherCATフレームからの情報が使用されます。検査結果が負の場合、次のEtherCATスレーブおよびEtherCATマスターのカウント値をインクリメントすることにより、EtherCATフレームに不良フラグが付けられます。
- ワーキングカウンタ: ワーキングカウンタはデータグラムの一部です。データ交換が正常に完了すると、EtherCATスレーブがカウンタ値を1つインクリメントします。EtherCATマスターは、戻されたEtherCATデータグラムのカウンタ値と予想されるカウンタ値を比較することで、データ交換エラーを検出します。

3.10.2.2 エラー応答

駆動機器はEtherCATマスターからの出力データを適宜受信し、機器の状態をEtherCATマスターに定期的に送信する必要があります。これは、EtherCATマスターからの出力データと駆動機器からの入力データを表示するプロセスイメージが、EtherCATマスターと駆動機器間で定期的に交換され、タイミングの過度な変動がなかった場合にのみ実行されます。

プロセスイメージの交換は、それぞれ異なる原理で動作する駆動機器の2つのメカニズムによって監視されます。

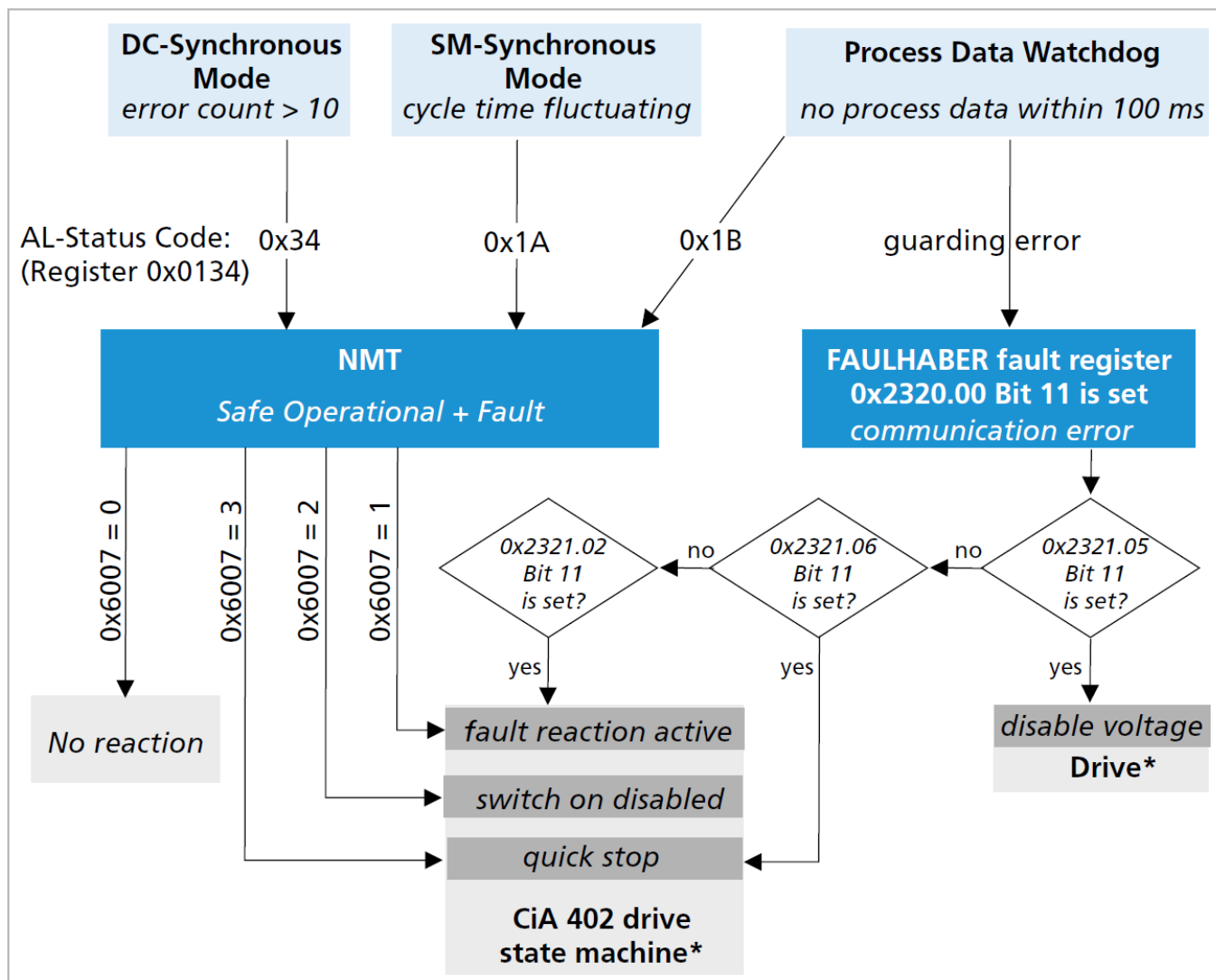


図7: エラー監視メカニズム

* CiA402に準拠した駆動機器プロファイルのオブジェクトは、ドライブ機能のマニュアルで説明されていません。

プロセスデータ到着時間の監視

プロセスデータが駆動機器に到着すると、到達時間が予想時間と一致するかチェックされます。時間の偏差が大き過ぎる状態が複数回検出されると、EtherCAT機器状態は安全運転およびエラー状態に切り替わります。オブジェクト0x6007に設定されたエラー応答によって、駆動機器は停止します（ドライブ機能のマニュアル参照）。

これらの監視は、同期の種類によって名前が異なります。

- DC同期: SYNC0監視
- SM同期: SyncManager監視

プロセスデータ受信の監視

プロセスデータを駆動機器が受け取れない場合（ケーブル断線などにより）は、到達時間の監視が有効になりません。このような状態でプロセスデータが100ms以上受信されない場合、プロセスデータウォッチドッグはエラー応答をトリガします。エラー応答は、FAULHABERエラーレジスタ0x2320.00のビット11のステータスによって異なります。

■ ビット11セットなし:

EtherCAT機器状態は安全運転およびエラー状態に切り替わります。

オブジェクト0x6007に設定されたエラー応答によって、駆動機器は停止します（ドライブ機能のマニュアル参照）。

■ ビット11セットあり:

駆動機器は、オブジェクト0x2321.02、0x2321.04、0x2321.05および0x2321.06に定義される通りに停止します（5.2節、48ページ参照）。

3.10.2.3 ネットワークトラフィック分析

ソフトウェアツール（Wiresharkなど）を使用してネットワークトラフィックを分析できます。ソフトウェアツールはネットワークに接続された別のPCにインストールすることも、EtherCATマスターに直接インストールすることもできます。ネットワークトラフィック分析では、EtherCATマスターから送信されたフレームとEtherCATマスターが受信したフレームを読み込んで比較します。エラー分析で特に重要なポイントは、上記のEtherCATフレームエントリ（FCS、ワーキングカウンタ）です。

3.10.2.4 EtherCAT AL Status Codeとトラブルシューティング

通信エラーが発生した場合、エラーコードがAL Status Codeレジスタ（0x0134）にロードされます。以下の表に、利用可能なコードとトラブルシューティングを記載しています。

AL Status Code	詳細	トラブルシューティング
0x0001	EtherCATシステム（ハードウェアまたはソフトウェア）が初期化しない。	FAULHABERサポートにお問い合わせください。
0x0002	SyncManager（マッピングまたはバッファ）のメモリーが利用できない。	FAULHABERサポートにお問い合わせください。
0x0011	要求したターゲット状態にならない／要求した遷移が許可されない。	AL機器状態を1つずつインクリメントしてください。 無効な手順の例 <ul style="list-style-type: none">■ 初期化 → 安全運転■ 初期化 → 運転■ 運転前 → 運転
0x0012	要求したAL機器状態のターゲット状態が存在しない。	以下のAL Status Codeのみが有効です。 <ul style="list-style-type: none">■ 初期化: 1■ 運転前: 2■ 安全運転: 4■ 運転: 8
0x0013	起動状態が実行されない。	起動状態を実行しようとししないでください。
0x0015	起動状態で正常に変更されない。メールボックス通信の構成（SDO通信など）が不良。	0x0016を参照してください。
0x0016	システムを運転前状態に切り替えたとき、メールボックス（SDO通信）の構成でエラーが検出された。	構成をチェックしてください。 <ul style="list-style-type: none">■ SM0およびSM1が有効か？■ SMの制御バイトは正しいか？■ SMの物理アドレスおよび設定アドレスは正しいか？■ SMの長さは制限範囲内か？

AL Status Code	詳細	トラブルシューティング
0x0017	SMのマッピングエントリが正しくない。	構成をチェックしてください。 <ul style="list-style-type: none"> マッピングされた全てのオブジェクトの長さがSMの長さを超えていないこと 最大8個のSMが使用可能なこと SMの重複禁止
0x001A	同期不良: 不良のしきい値 > 0で、内部エラーカウンタがこの値を超えた。 プロセスデータの送信が早過ぎたまたは遅過ぎた。 SM同期モードのみ。	プロセスデータをより正確に送信します。 または 0x1C32.02に設定されたサイクル時間を確認します。実際のサイクル時間と一致している必要があります。
0x001B	同期不良: プロセスデータを100ms以上受信していないため、SMウォッチドッグが通信不良を報告した。	プロセスデータを送信してください。
0x001D	出力SMの間違った構成。	SMの正しい構成を確認してください。 <ul style="list-style-type: none"> 長さが0のSMを無効にし、他の全てを有効にする。 SMの正しい制御バイトを使用する。 SMの物理アドレスおよび設定アドレスは正しいか？ SMの長さは制限範囲内か？ SMの重複禁止
0x001E	入力SMの間違った構成。	SMの正しい構成を確認してください。 <ul style="list-style-type: none"> 長さが0のSMを無効にし、他の全てを有効にする。 SMの正しい制御バイトを使用する。 SMの物理アドレスおよび設定アドレスは正しいか？ SMの長さは制限範囲内か？ SMの重複禁止
0x0026	SyncManagerの設定の不一致。	設定を確認してください。 以下の設定の組み合わせのみが有効です。 <ul style="list-style-type: none"> SM0 = 0 & SM1 = 1 (Freerun) SM0 = 1 & SM1 = 0x22 & ADO 0981 (SyncOut Unit) Bits [3,1,0] = 000 (SYNC0非アクティブ) SM0 = 2 & SM1 = 2 & ADO 0981 (SyncOut Unit) Bits [3,1,0] = 011 (SYNC0アクティブ)
0x002C	DCモード時にSYNC0信号がない。	SYNC0の構成を確認してください。
0x002E	サイクル時間が短過ぎる。	FreerunおよびSM同期モードの最小サイクル時間は1msです。
0x0030	DCモードの間違った構成。	DCの正しい構成を確認してください。 <ul style="list-style-type: none"> SM0 = 0x02 SM1 = 0x02 ADO 0981 (SyncOut Unit) Bits [3,1,0] = 011 (SYNC0アクティブ)
0x0034	DCステータス時に、利用できるプロセスイメージがないため頻繁に出力しなければならぬ。 DCモードのみ。	SYNC0パルスにตอบสนองして使用できるように、プロセスイメージを正しい時間に送信します。
0x0036	DC時間が短過ぎる。	最小DC時間は500 μ sです。
0xB001	SyncManagerチャンネルでPDOが確認できない。	FAULHABERサポートにお問い合わせください。
0xB003	PDOがマッピングされていない。	FAULHABERサポートにお問い合わせください。

AL Status Code	詳細	トラブルシューティング
0xB004	PDOインデックスが必要な範囲にない。	FAULHABERサポートにお問い合わせください。
0xB007	オブジェクトディクショナリにPDOエントリがない。	FAULHABERサポートにお問い合わせください。

3.11 パラメータの保存および復元

コントローラの電源を再度投入した後も、ODで変更されたパラメータが保持されるように不揮発性メモリー（アプリケーションEEPROM）に恒久的に保存するには、「Save」コマンドを実行します（5.1節、40ページ参照）。モータの電源を投入すると、パラメータが不揮発性メモリーから揮発性メモリー（RAM）に自動的にロードされます。

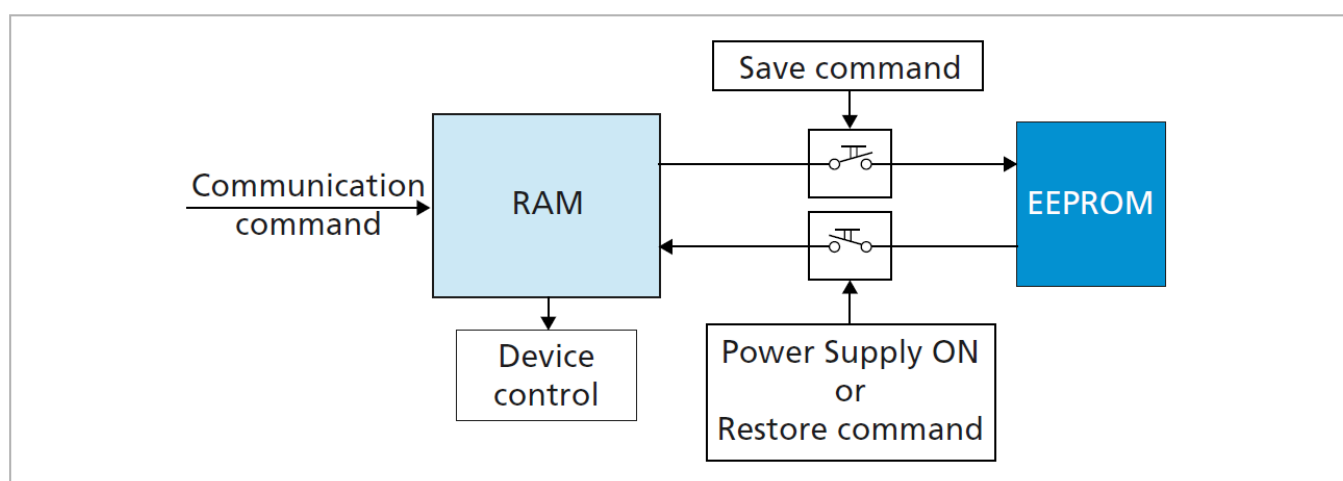


図8: パラメータの保存および復元

「Restore」コマンドを使用すると、以下のパラメータをロードできます（5.1節、40ページ参照）。

- 初期設定（工場出荷時の設定）
- 「Save」コマンドを使用して保存したパラメータ

3.11.1 パラメータの保存

現在のパラメータ設定は、全体または一部を内部EEPROM（SAVE）に保存できます（表14参照）。

- ▶ 「Save」をオブジェクト0x1010のサブインデックス01～05に書き込みます（表15参照）。

3.11.2 設定の復元

i 次に駆動機器の電源を投入したときに、保存されたパラメータは自動的にロードされます。

初期設定または最後に保存したパラメータ設定は、全体または一部を内部EEPROMからいつでもロードすることができます（RESTORE）（表16参照）。

- 「Load」署名をオブジェクト0x1011のサブインデックス01～06に書き込みます（表17参照）。
 - Restore Factory（01）、Restore Communication（02）およびRestore Application（03）の後に、リセットするとパラメータが更新されます。
- 「Reload」コマンドで、アプリケーションパラメータ（04）と特別なアプリケーションパラメータ（05/06）のレコード1およびレコード2と一緒に更新できます。
 - 「Reload」コマンドは、アプリケーションパラメータとして最後に保存した値を上書きします。

i 現在ロードしている値を復元した後に使用する場合は、適切なプログラム（FAULHABER Motion Managerなど）を使用してPCに保存する必要があります。

3.11.3 パラメータセットの変更

保存したアプリケーションパラメータ（モータデータ、I/O構成、コントローラパラメータ、動作モードなど）には総合的なパラメータの基本セット（App）が含まれています。また、ロード状況に応じて、頻繁に修正が必要なパラメータを保存するデータエリア（App1/App2）もあります。

速度コントローラおよびフィルタ

インデックス	サブインデックス	名前	型	属性	意味
0x2344	0x01	Gain KP,n [As 1e-6]	U32	rw	コントローラのゲイン
	0x02	Integral time TN.n [100 μs]	U16	rw	コントローラリセット時間
0x2346	0x01	Setpoint velocity filter time T_F [100 μs]	U16	rw	フィルタ時間T_F
	0x02	Setpoint filter enable	U8	rw	0: 非アクティブ状態 1: アクティブ状態
	0x01	Gain factor (applied on KP of velocity controller in PP mode)	U8	rw	ゲイン係数（速度コントローラによりKPのPPモードで使用） 0: 速度コントローラのゲイン係数はターゲットで0に減速する 128: 可変ゲインなし 255: 速度コントローラのゲイン係数はターゲットで2倍になる

位置コントローラ

インデックス	サブインデックス	名前	型	属性	意味
0x2348	0x00	Number of entries	U8	ro	オブジェクトエントリの数
	0x01	K _v [1/s]	U8	rw	範囲: 1～250

プリ制御

インデックス	サブインデックス	名前	型	属性	意味
0x2349	0x01	Torque/force feedforward factor	U8	rw	トルク／力の制御の係数 0: プリ制御値の0%アクティブ化 128: 100%プリ制御
	0x02	Torque/Force feed forward delay	U8	rw	設定点遅延: 0: 遅延のないアクティブ化 1: 1つのサンプリングによるアクティブ化遅延
0x234A	0x01	Velocity feed forward factor	U8	rw	トルク／力の制御の係数 0: 0%プリ制御 128: 100%プリ制御
	0x02	Velocity feed forward delay	U8	rw	設定点遅延 : 0: 遅延のないアクティブ化 1: 1つのサンプリングによるアクティブ化遅延

全般設定

インデックス	サブインデックス	名前	型	属性	意味
0x6060	0x00	Modes of operation	S8	rw	動作モードの選択 -4: ATC -3: AVC -2: APC -1: 電圧モード 0: コントローラ非アクティブ 1: PP 3: PV 6: 原点復帰 8: CSP 9: CSV 10: CST
0x6081	0x00	Profile Velocity	U32	rw	プロファイル速度[ユーザ定義のスケーリング]
0x6083	0x00	Profile acceleration	U32	rw	プロファイル加速[1/s ²]
0x6084	0x00	Profile deceleration	U32	rw	プロファイル減速[1/s ²]
0x6086	0x00	Motion profile type	S16	rw	モーションプロファイルの種類: 0: リニアプロファイル 1: Sin ² 速度
0x60E0	0x00	Positive torque limit value	S16	rw	上限値[相対的なスケーリング]
0x60E1	0x00	Negative torque limit value	S16	rw	下限値[相対的なスケーリング]

これらのパラメータは2回保存されます。動作時にシステムは、プリセット値を素早く切り替えることができます。

アプリケーションの作成


- ▶ アプリケーションパラメータ1の保存 「Save」 署名をオブジェクト0x1010のサブインデックス04に書き込みます。
 - ↳ 現在のデータが、アプリケーションパラメータセット1として保存されます。
- ▶ アプリケーションパラメータ2の保存 「Save」 署名をオブジェクト0x1010のサブインデックス05に書き込みます。
 - ↳ 現在のデータが、アプリケーションパラメータセット2として保存されます。

アプリケーションのアクティブ化

- ▶ アプリケーションパラメータ1のリロード: 「Load」 署名をオブジェクト0x1011のサブインデックス05に書き込みます。
 - ↳ アプリケーションパラメータセット1の現在のデータが直接アクティブ化されます。
- ▶ アプリケーションパラメータ2のリロード: 「Load」 署名をオブジェクト0x1011のサブインデックス06に書き込みます。
 - ↳ アプリケーションパラメータセット2の現在のデータが直接アクティブ化されます。

4 追跡レコーダ

追跡レコーダを使用すると、コントローラのパラメータを4個まで記録できます。オブジェクトディクショナリには、トリガソースが収録されています。最大4個の信号ソースが選択できます。パラメータ値が内部バッファに書き込まれ、その後読み込むことができます（4.2節、38ページ参照）。プロセスデータの周期的送信と比較し、高速で処理できるメリットがあります。追跡レコーダは、コントローラのデータを100 μ sのサンプリング間隔で記録できるのに対し、プロセスデータは500 μ sの間隔でのみ送信できます。

 FAULHABER Motion Managerを使用すると、追跡機能を簡単に設定、評価できます。

SDOを使用して追跡レコーダのデータの設定および読み込みを実行します。

追跡レコーダは、ODのオブジェクト0x2370を使用して設定します。

セグメント化SDOアップロードプロトコルを使用して、記録されたデータを読み込みます。オブジェクト0x2371はこの目的のためにODで使用できます（4.2節、38ページ参照）。

4.1 追跡設定

追跡レコーダの設定にはオブジェクト0x2370を使用します。ここから、記録するデータソース、バッファサイズ、分解能、トリガ条件を設定できます。

表10: 追跡設定（0x2370）

インデックス	サブインデックス	名前	型	属性	デフォルト値	意味
0x2370	0x00	Number of entries	U8	ro	10	オブジェクトエントリの数
	0x01	Trigger value	U32	wo	0	トリガ値
	0x02	Trigger threshold	S32	rw	0	トリガしきい値
	0x03	Trigger offset (delay)	S16	rw	0	トリガ遅延
	0x04	Trigger mode	U16	rw	0	トリガモード
	0x05	Buffer length	U16	rw	100	バッファ長
	0x06	Sample time	U8	rw	1	サンプリングレートの記録 1: 各サンプリング手順
	0x07	Trace source of channel 1	U32	wo	0	チャンネル1の追跡ソース
	0x08	Trace source of channel 2	U32	wo	0	チャンネル2の追跡ソース
	0x09	Trace source of channel 3	U32	wo	0	チャンネル3の追跡ソース
	0x0A	Trace source of channel 4	U32	wo	0	チャンネル4の追跡ソース

トリガソース（0x2370.01）、ソース1~4（0x2370.07~0A）

記録されるパラメータ（ソース1~ソース4）は、対応するオブジェクトエントリ（必要なパラメータのインデックスとサブインデックス）へのポインタとして、オブジェクト0x2370.07~0x2370.0Aに入力する必要があります。トリガソースは、オブジェクト0x2370.01に対応するオブジェクトエントリ（目的のパラメータのインデックスとサブインデックス）へのポインタとして入力する必要があります。

例:

最初のデータソースとして、オブジェクト0x6064.00（位置の実際の値）を記録する必要があり、値0x606400をオブジェクト0x2370.07に入力する必要があります。

トリガしきい値 (0x2370.02)

トリガしきい値がオブジェクト0x2370.02に入力されます。

トリガの種類オブジェクト0x2370.04のビット1~3の設定に応じて、ここでセットされた上下限のしきい値から記録が開始されます。

トリガ遅延 (0x2370.03)

トリガ遅延は、オブジェクト0x2370.06にセットされたサンプリング時間の倍数でオブジェクト0x2370.03に記述されています。

- 遅延 > 0: サンプリング時間のセットされた倍数に定義された時間に記録が開始されます。
- 遅延 < 0: 負の遅延は、バッファの長さまで実行できます。現在のトリガの記録は、リングバッファの記録が開始された時点で終了します。これによりトリガの前に記録された値が維持されます。

トリガモード (0x2370.04)

トリガおよびデータソースの種類は、オブジェクト0x2370.04によって指定されます。ビット0はトリガをアクティブにし、トリガ条件を満たしていれば記録を開始します。

表11: トリガモード (0x2370.04)

ビット	エントリ	詳細
0 (LSB)	EN	<ul style="list-style-type: none">0: アクティブなトリガなし1: アクティブなトリガはトリガモード1および3で自動でリセット
1	Edge 0	<ul style="list-style-type: none">0: 立上りまたはトリガ > しきい値
2	Edge 1	<ul style="list-style-type: none">1: 立下げまたはトリガ < しきい値
3	Edge 2	
4 to 5	Reserved	-
6	Mode 0	<ul style="list-style-type: none">0: トリガなし
7	Mode 1	<ul style="list-style-type: none">1: 単一ショット2: 繰り返し
8 to 10	Reserved	-
11	Source type 1	<ul style="list-style-type: none">0: オブジェクトディクショナリエントリをソースとして使用
12	Source type 2	<ul style="list-style-type: none">1: 現在サポートされていない
13	Source type 3	
14	Source type 4	
15 (MSB)	Trigger type	

バッファサイズ (0x2370.05)

記録可能なバッファ長はオブジェクト0x2370.05にセットされていますが、許容される長さは記録するパラメータのデータの種類によって異なります。データソースあたり最大2kBのバッファが利用可能です。

サンプリング時間 (0x2370.06)

サンプリングレートは、コントローラのサンプリング時間の倍数としてオブジェクト0x2370.06に記述されています。


4.2 追跡バッファの読み込み

記録されたデータバッファは、オブジェクト0x2371を使用して読み込むことができます。

表12: 追跡バッファ (0x2371)

インデックス	サブインデックス	名前	型	属性	デフォルト値	意味
0x2371	0x00	Number of entries	U8	ro	5	オブジェクトエントリの数
	0x01	Trace state	U16	ro	0	追跡状態
	0x02	Trace value of channel 1	Vis string	ro	-	信号バッファ、チャンネル1
	0x03	Trace value of channel 2	Vis string	ro	-	信号バッファ、チャンネル2
	0x04	Trace value of channel 3	Vis string	ro	-	信号バッファ、チャンネル3
	0x05	Trace value of channel 4	Vis string	ro	-	信号バッファ、チャンネル4

個別のデータソースのユーザデータ長は、送信するパラメータのデータ長（ODエントリによる）およびセットされたバッファサイズにより異なるため、記録された値の読み込みには、各データソースごとにデータ長とバッファサイズを掛けたメモリー容量が必要になります。

 個別のデータポイントは、追跡レコーダの最も高い分解能で記録できます。

トリガステータス (0x2371.01)

表13: トリガステータス (0x2371.01)

ビット	エントリ	詳細	
0 (LSB)	Status 0	<ul style="list-style-type: none"> 0: アクティブなトリガなし 1: トリガに達していない 2: 記録の未完了 3: 記録は完了し、データ利用可 	
1	Status 1		
2 to 7	not used		-
8 to 15 (MSB)	Start index		トリガ後のバッファの最初の値

記録データを読み込む前に、トリガステータス0x2371.01を確認する必要があります。ビット0またはビット1がセットされている場合（ステータス = 3）は記録が完了します。バッファの内容は、セグメント化SDOアップロードプロトコルでオブジェクト0x2371.02~0x2371.05を使用して読み込むことができます。

4.3 追跡機能の一般的な実行

1. トリガおよびデータソースの種類 (2370.04) を設定します。
2. トリガソースおよび記録する信号 (2370.01、07~0A) を設定します。
3. 記録長 (2370.05) を設定します。
4. 必要な場合、サンプリングレート (2370.06) を設定します。
5. トリガのしきい値 (2370.02) を設定します。
6. トリガの立上り／立下りを設定し、記録 (2370.04) を有効にします。
👉 以上で追跡レコーダの設定は完了です。
7. 値3のトリガステータス (2371.01) を設定します。
8. バッファの記録内容 (2371.02 to 05) を読み込みます。

5 パラメータの説明

5.1 CiA 301の通信オブジェクト

機器の種類

インデックス	サブインデックス	名前	型	属性	デフォルト値	意味
0x1000	0x00	Device type	U32	ro	0x00420192	機器の種類を表示

2つの16ビットフィールドでコード化された機器の種類の情報を含む。

- バイトMSB（最上位バイト）：追加情報 = 0x192 (402d)
- バイトLSB（最下位バイト）：0x42（サーボドライブ、型固有のPDOマッピング）

エラーレジスタ

インデックス	サブインデックス	名前	型	属性	デフォルト値	意味
0x1001	0x00	Error register	U8	ro	Yes	エラーレジスタ

エラーレジスタは、ビットコード化された最新のエラーの記録を含みます。
このパラメータはPDOにマッピングできます。

事前定義されたエラーフィールド（エラーログ）

インデックス	サブインデックス	名前	型	属性	デフォルト値	意味
0x1003	0x00	Number of errors	U8	rw	-	保存されたエラー数
	0x01–0x08	Standard error field	U32	ro	-	最新のエラーコード

エラーログには、最新のエラーのコードが含まれています。

- バイトMSB: エラーレジスタ
- バイトLSB: エラーコード

エラーコードの意味は、3.6節、20ページを参照してください。

サブインデックス0に0を書き込むと、エラーログが消去されます。

メーカーのデバイス名

インデックス	サブインデックス	名前	型	属性	デフォルト値	意味
0x1008	0x00	Manufacturer's device name	Vis string	const	-	デバイス名

メーカーのハードウェアバージョン

インデックス	サブインデックス	名前	型	属性	デフォルト値	意味
0x1009	0x00	Manufacturer's hardware version	Vis string	const	-	ハードウェアバージョン

メーカーのソフトウェアバージョン

インデックス	サブインデックス	名前	型	属性	デフォルト値	意味
0x100A	0x00	Manufacturer's software version	Vis string	const	-	ソフトウェアバージョン

パラメータの保存

表14: パラメータの保存


インデックス	サブインデックス	名前	型	属性	デフォルト値	意味
0x1010	0x00	Number of entries	U8	ro	9	オブジェクトエントリの数
	0x01	Save all parameters	U32	rw	1	全てのパラメータの保存
	0x02	Save comm parameters	U32	rw	1	通信パラメータの保存 (オブジェクトデイクシヨナリエントリ0x0000~0x1FFF)
	0x03	Save app parameters	U32	rw	1	アプリケーションパラメータの保存 (オブジェクトデイクシヨナリエントリ0x2000~0x6FFF)
	0x04	Save app parameters 1	U32	rw	1	直接変更のためアプリケーションパラメータの保存 (セット1)
	0x05	Save app parameters 2	U32	rw	1	直接変更のためアプリケーションパラメータの保存 (セット2)

「パラメータの保存」オブジェクトは設定パラメータをフラッシュメモリーに保存します。読み込みアクセスで保存オプションの情報が提供されます。「Save」署名を各サブインデックスに書き込むと、保存手順が開始されます。

表15: 「Save」署名

署名	ISO 8859 (ASCII)	16進法
MSB	e	65 h
	v	76 h
	a	61 h
LSB	s	73 h

注意!

 フラッシュメモリーは10,000回の書き込みサイクルに対応するように設計されています。このコマンドを10回以上実行すると、フラッシュメモリーの正常な動作が保証されなくなります。

- ▶ 頻繁に保存しないでください。
- ▶ 10,000回保存した後は、機器を交換してください。

パラメータ初期設定の復元

表16: パラメータの復元


インデックス	サブインデックス	名前	型	属性	デフォルト値	意味
0x1011	0x00	Number of entries	U8	ro	6	オブジェクトエントリの数
	0x01	Restore all default parameters	U32	rw	1	全ての初期設定の復元
	0x02	Restore comm default parameters	U32	rw	1	通信パラメータの全ての初期設定の復元 (0x0000~0x1FFF)
	0x03	Restore app default parameters	U32	rw	1	アプリケーションパラメータの全ての初期設定の復元 (~0x2000)
	0x04	Reload user parameters	U32	rw	1	ユーザにより最後に保存されたアプリケーションパラメータの復元 (~0x2000)


インデックス	サブインデックス	名前	型	属性	デフォルト値	意味
	0x05	Reload application parameters 1	U32	rw	1	直接変更のためのアプリケーションパラメータセット1
	0x06	Reload application parameters 2	U32	rw	1	直接変更のためのアプリケーションパラメータセット2

「パラメータ初期設定の復元」オブジェクトは標準設定パラメータをロードします。標準設定パラメータには、初期パラメータと最後に保存されたパラメータがあります。読み込みアクセスで復元オプションの情報が提供されます。「Load」署名を各サブインデックスに書き込むと、復元手順が実行されます。

表17: 「Load」署名

署名	ISO 8859 (ASCII)	16進法
MSB	d	64 h
	a	61 h
	o	6Fh
LSB	l	6Ch

 出力状態は、出カステージがオフになっているときにのみロードされます。

 『Restore Factory Settings』で復元されたパラメータを有効にするには、駆動機器の電源を入れ直す必要があります。

識別オブジェクト

インデックス	サブインデックス	名前	型	属性	デフォルト値	意味
0x1018	0x00	Number of entries	U8	ro	4	オブジェクトエントリの数
	0x01	Vendor ID	U32	ro	327	メーカーのコード番号 (FAULHABER: 327)
	0x02	Product code	U32	ro	48	製品コード番号
	0x03	Revision number	U32	ro	-	バージョン番号
	0x04	Serial number	U32	ro	-	シリアル番号

PDO1マッピング受信

インデックス	サブインデックス	名前	型	属性	デフォルト値	意味
0x1600	0x00	Number of mapped objects	U8	ro	1	マッピングされたオブジェクトの数
	0x01	RxPDO1 mapping entry 1	U32	rw	0x60400010	16ビットの制御ワード (0x6040) へのポインタ
	0x02	RxPDO1 mapping entry 2	U32	rw	0	
	0x03	RxPDO1 mapping entry 3	U32	rw	0	
	0x04	RxPDO1 mapping entry 4	U32	rw	0	

PDO2マッピング受信

インデックス	サブインデックス	名前	型	属性	デフォルト値	意味
0x1601	0x00	Number of mapped objects	U8	ro	2	マッピングされたオブジェクトの数
	0x01	RxPDO2 mapping entry 1	U32	rw	0x60400010	16ビットの制御ワード (0x6040) へのポインタ
	0x02	RxPDO2 mapping entry 2	U32	rw	0x607A0020	32ビットの目標位置 (0x607A) へのポインタ
	0x03	RxPDO2 mapping entry 3	U32	rw	0	
	0x04	RxPDO2 mapping entry 4	U32	rw	0	

PDO3マッピング受信

インデックス	サブインデックス	名前	型	属性	デフォルト値	意味
0x1602	0x00	Number of mapped objects	U8	ro	2	マッピングされたオブジェクトの数
	0x01	RxPDO3 mapping entry 1	U32	rw	0x60400010	16ビットの制御ワード (0x6040) へのポインタ
	0x02	RxPDO3 mapping entry 2	U32	rw	0x60FF0020	32ビットの目標速度 (0x60FF) へのポインタ
	0x03	RxPDO3 mapping entry 3	U32	rw	0	
	0x04	RxPDO3 mapping entry 4	U32	rw	0	

PDO4マッピング受信

インデックス	サブインデックス	名前	型	属性	デフォルト値	意味
0x1603	0x00	Number of mapped objects	U8	ro	2	マッピングされたオブジェクトの数
	0x01	RxPDO4 mapping entry 1	U32	rw	0x60400010	16ビットの制御ワード (0x6040) へのポインタ
	0x02	RxPDO4 mapping entry 2	U32	rw	0x60710010	16ビットの目標トルク (0x6071) へのポインタ
	0x03	RxPDO4 mapping entry 3	U32	rw	0	
	0x04	RxPDO4 mapping entry 4	U32	rw	0	

PDO1マッピング送信

インデックス	サブインデックス	名前	型	属性	デフォルト値	意味
0x1A00	0x00	Number of mapped objects	U8	rw	1	マッピングされたオブジェクトの数
	0x01	TxPDO1 mapping entry 1	U32	rw	0x60410010	16ビットの状態ワード (0x6041) へのポインタ
	0x02	TxPDO1 mapping entry 2	U32	rw	0	
	0x03	TxPDO1 mapping entry 3	U32	rw	0	
	0x04	TxPDO1 mapping entry 4	U32	rw	0	

PDO2マッピング送信

インデックス	サブインデックス	名前	型	属性	デフォルト値	意味
0x1A01	0x00	Number of mapped objects	U8	rw	2	マッピングされたオブジェクトの数
	0x01	TxPDO2 mapping entry 1	U32	rw	0x60410010	16ビットの状態ワード (0x6041) へのポインタ
	0x02	TxPDO2 mapping entry 2	U32	rw	0x60640020	32ビットの実位置値 (0x6064) へのポインタ
	0x03	TxPDO2 mapping entry 3	U32	rw	0	
	0x04	TxPDO2 mapping entry 4	U32	rw	0	

PDO3マッピング送信

インデックス	サブインデックス	名前	型	属性	デフォルト値	意味
0x1A02	0x00	Number of mapped objects	U8	rw	2	マッピングされたオブジェクトの数
	0x01	TxPDO3 mapping entry 1	U32	rw	0x60410010	16ビットの状態ワード (0x6041) へのポインタ
	0x02	TxPDO3 mapping entry 2	U32	rw	0x606C0020	32ビットの実速度値 (0x606C) へのポインタ
	0x03	TxPDO3 mapping entry 3	U32	rw	0	
	0x04	TxPDO3 mapping entry 4	U32	rw	0	

PDO4マッピング送信

インデックス	サブインデックス	名前	型	属性	デフォルト値	意味
0x1A03	0x00	Number of mapped objects	U8	rw	2	マッピングされたオブジェクトの数
	0x01	TxPDO4 mapping entry 1	U32	rw	0x60410010	32ビットの実位置値 (0x6064) へのポインタ
	0x02	TxPDO4 mapping entry 2	U32	rw	0x60770010	16ビットの実トルク値 (0x6077) へのポインタ
	0x03	TxPDO4 mapping entry 3	U32	rw	0	
	0x04	TxPDO4 mapping entry 4	U32	rw	0	

SyncManager通信の種類

インデックス	サブインデックス	名前	型	属性	デフォルト値	意味
0x1C00	0x00	Number of objects	U8	ro	4	オブジェクトの数
	0x01	SM1 communication type	U8	ro	0	0: SyncManager未使用 1: メールボックス受信 (マスターからスレーブ)
	0x02	SM2 communication type	U8	ro	0	2: メールボックス送信 (スレーブからマスター)
	0x03	SM3 communication type	U8	ro	0	3: プロセスデータ出力 (マスターからスレーブ)
						4: プロセスデータ入力 (スレーブからマスター)
0x04	SM4 communication type	U8	ro	0		

SyncManager 2 (RxPDO、マスターから駆動機器) : PDOマッピング

インデックス	サブインデックス	名前	型	属性	デフォルト値	意味
0x1C12	0x00	Number of objects	U8	rw	4	オブジェクトの数
	0x01	SM2: 1st RxPDO assignment	U16	rw	0x1600	SyncManagerチャンネル2の受信PDO 1への割り当て 利用可能な値: 0x1600...0x1603
	0x02	SM2: 2nd RxPDO assignment	U16	rw	0x1601	SyncManagerチャンネル2の受信PDO 2への割り当て 利用可能な値: 0x1600...0x1603
	0x03	SM2: 3rd RxPDO assignment	U16	rw	0x1602	SyncManagerチャンネル2の受信PDO 3への割り当て 利用可能な値: 0x1600...0x1603
	0x04	SM2: 4th RxPDO assignment	U16	rw	0x1603	SyncManagerチャンネル2の受信PDO 4への割り当て 利用可能な値: 0x1600...0x1603

SyncManager 3 (TxPDO、駆動機器からマスター) : PDOマッピング

インデックス	サブインデックス	名前	型	属性	デフォルト値	意味
0x1C13	0x00	Number of objects	U8	rw	4	オブジェクトの数
	0x01	SM3: 1st TxPDO assignment	U16	rw	0x1A00	SyncManagerチャンネル3の送信PDO 1への割り当て 利用可能な値: 0x1A00...0x1A03
	0x02	SM3: 2nd TxPDO assignment	U16	rw	0x1A01	SyncManagerチャンネル3の送信PDO 2への割り当て 利用可能な値: 0x1A00...0x1A03
	0x03	SM3: 3rd TxPDO assignment	U16	rw	0x1A02	SyncManagerチャンネル3の送信PDO 3への割り当て 利用可能な値: 0x1A00...0x1A03
	0x04	SM3: 4th TxPDO assignment	U16	rw	0x1A03	SyncManagerチャンネル3の送信PDO 4への割り当て 利用可能な値: 0x1A00...0x1A03

SyncManager 2 (RxPDO、マスターから駆動機器) : パラメータ

インデックス	サブインデックス	名前	型	属性	デフォルト値	意味
0x1C32	0x00	Number of objects	U8	ro	12	入力PDO用Sync-Managerのパラメータ
	0x01	SM2: Synchronisation type	U16	rw	1	1 同期の種類: <ul style="list-style-type: none"> ▪ 0: FreeRun ▪ 1: SM-Sync ▪ 2: DC-Sync
	0x02	SM2: Cycle time	U32	rw	500000	サイクル時間 (値は必ず500000nsの倍数)
	0x04	SM2: Synchronisation types supported	U16	ro	0	サポートされた同期の種類
	0x05	SM2: Minimum cycle time	U32	ro	0	最小サイクル時間 (DC-Syncモードのみ)
	0x06	SM2: Calc and copy time	U32	ro	0	次のSyncManagerイベントが到着可能な最短時間 (ns) (DC-Syncモードのみ)
	0x09	SM2: Delay time	U32	ro	0	出力するまでのハードウェア遅延時間 (DC-Syncモードのみ)
	0x0B	SM2: SM event missed counter	U16	ro	0	喪失したSMイベント数 (DC-Syncモードのみ)
	0x0C	SM2: Cycle time too short counter	U16	ro	0	次のSM2イベントが到着する前に、プロセス入力データが更新されないとエラーカウンタは1にインクリメントされます。(FreeRunモード以外)

SyncManager 3 (TxPDO、駆動機器からマスター) : パラメータ

インデックス	サブインデックス	名前	型	属性	デフォルト値	意味
0x1C33	0x00	SyncManager 3 (TxPDO): Parameters	U8	ro	12	PDOを送信するSyncManagerのパラメータ
	0x01	SM3: Synchronisation type	U16	rw	34	同期の種類: <ul style="list-style-type: none"> ▪ 0: FreeRun ▪ 2: SM-Sync ▪ 34: DC-Sync~SM2
	0x02	SM3: Cycle time	U32	ro	0	0x1C32.02の値のコピー 出力が定義されず入力のみの場合、1C33.02を設定する必要があります。この場合、1C32.02は設定できません。通常（入出力がともに定義される場合）は、1C32.02および1C33.02が定義されますが、両方とも内部では同じ変数を指します。これにより、同じ時間のみ使用可能にします。
	0x04	SM3: Synchronisation types supported	U16	ro	0	サポートされた同期の種類
	0x05	SM3: Minimum cycle time	U32	ro	0	最小サイクル時間（DC-Syncモードのみ）
	0x06	SM3: Calc and copy time	U32	ro	0	入力の読み込みからマスターで入力が可能になるまでの時間（ns）（DC-Syncモードのみ）
	0x0B	SM3: SM event missed counter	U16	ro	0	喪失したSMイベント数（DC-Syncモードのみ）
	0x0C	SM3: Cycle time too short counter	U16	ro	0	次のSM2イベントが到着する前に、プロセス入力データが更新されないとエラーカウンタは1にインクリメントされます。（FreeRunモード以外）

5.2 メーカー固有のオブジェクト

FAULHABERエラーレジスタ

インデックス	サブインデックス	名前	型	属性	デフォルト値	意味
0x2320	0x00	Fault register	U16	ro	-	FAULHABERエラーレジスタ

エラーマスク

インデックス	サブインデックス	名前	型	属性	デフォルト値	意味
0x2321	0x00	Number of entries	U8	ro	6	オブジェクトエントリの数
	0x01	Emergency mask	U16	rw	0x00FF	エラーメッセージが送信されたエラー
	0x02	Fault mask	U16	rw	0x0000	駆動機器のマシンステータスを <i>Fault Reaction Active</i> ステータスに切り替えるエラー
	0x03	Error Out mask	U16	rw	0x00FF	エラー出力ピンがセットされたエラー
	0x04	Disable voltage mask	U16	ro	0x0000	駆動機器をオフにするエラー（設定不可）
	0x05	Disable voltage user mask	U16	rw	0x0000	駆動機器をオフにするエラー（設定可）
	0x06	Quick stop mask	U16	rw	0x0000	駆動機器のマシンステータスを <i>Quick Stop Active</i> ステータスに切り替えるエラー

駆動機器機器状態のステータスは、ドライブ機能のマニュアルで説明されています。

追跡設定

インデックス	サブインデックス	名前	型	属性	デフォルト値	意味
0x2370	0x00	Number of entries	U8	ro	10	オブジェクトエントリの数
	0x01	Trigger value	U32	wo	0	トリガ値
	0x02	Trigger threshold	S32	rw	0	トリガしきい値
	0x03	Trigger offset (delay)	S16	rw	0	トリガ遅延
	0x04	Trigger mode	U16	rw	0	トリガモード
	0x05	Buffer length	U16	rw	100	バッファ長
	0x06	Sample time	U8	rw	1	サンプリングレートの記録 1: 各サンプリング手順
	0x07	Trace source of channel 1	U32	wo	0	チャンネル1の追跡ソース
	0x08	Trace source of channel 2	U32	wo	0	チャンネル2の追跡ソース
	0x09	Trace source of channel 3	U32	wo	0	チャンネル3の追跡ソース
	0x0A	Trace source of channel 4	U32	wo	0	チャンネル4の追跡ソース

追跡バッファ

インデックス	サブインデックス	名前	型	属性	デフォルト値	意味
0x2371	0x00	Number of entries	U8	ro	5	オブジェクトエントリの数
	0x01	Trace state	U16	ro	0	追跡状態
	0x02	Trace value of channel 1	Vis string	ro	-	信号バッファ、チャンネル1
	0x03	Trace value of channel 2	Vis string	ro	-	信号バッファ、チャンネル2
	0x04	Trace value of channel 3	Vis string	ro	-	信号バッファ、チャンネル3
	0x05	Trace value of channel 4	Vis string	ro	-	信号バッファ、チャンネル4

RS232ボーレートインデックスおよびノード番号

インデックス	サブインデックス	名前	型	属性	デフォルト値	意味
0x2400	0x00	Number of entries	U8	rw	8	オブジェクトエントリの数
	0x02	RS232 rate	U8	rw	3	ボーレートインデックス
	0x03	Node ID	U8	rw	1	ノード番号
	0x08	Explicit Device ID	U16	rw	0	駆動機器の識別

〒140-0013

東京都 品川区 南大井 6-20-8

ユニゾ大森ビル 8F

新光電子株式会社

TEL. 03-6404-1003

FAX. 03-6404-1005

e-mail.motor-info@shinkoh-elecs.co.jp

www.shinkoh-faulhaber.jp

7000.05051, English, 2nd edition, 9-12-2016

© DR.FRITZ FAULHABER GMBH & CO.KG

仕様は予告なしに変更されることがあります。

DR.FRITZ FAULHABER

GMBH & CO.KG