

FAULHABER

通信マニュアル

MC 5010

MC 5005

MC 5004

MCS

CANopen®

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	CANopenデバイスの基本配置	8
2.2	通信の前提条件	9
2.3	FAULHABER Motion Manager	9
2.4	パラメータの保存および復元	10
2.4.1	パラメータの保存	10
2.4.2	設定の復元	11
2.4.3	パラメータセットの変更	11
3	CANopenプロトコルの説明	14
3.1	はじめに	14
3.2	通信サービス	15
3.3	識別子の配布	16
3.4	PDO（プロセスデータオブジェクト）	17
3.4.1	PDOの構造	17
3.4.2	標準構成のPDOマッピング（初期設定の状態）	18
3.4.3	マッピングエラーの取り扱い	19
3.4.4	ダミーマッピング	20
3.5	SDO（サービスデータオブジェクト）	21
3.5.1	優先転送	21
3.5.2	SDOエラーの詳細	22
3.6	緊急オブジェクト（エラーメッセージ）	23
3.7	SYNCオブジェクト	26
3.7.1	同期PDOのトリガ	26
3.8	NMT（ネットワーク管理）	27
3.8.1	起動	29
3.8.2	監視機能	29
3.8.2.1	ノードガード	29
3.8.2.2	ハートビート	31
3.8.3	監視機能の設定	32
3.9	オブジェクトディクショナリのエントリ	32
3.10	エラーハンドリング	33
3.10.1	CANエラー	33
3.10.2	機器不良	33

4	追跡	35
4.1	追跡レコーダ	35
4.1.1	追跡設定	35
4.1.2	追跡バッファの読み込み	37
4.1.3	追跡機能の一般的な実行	38
4.2	追跡ロガー	38
5	通信設定	39
5.1	CANネットワーク経由の設定	39
5.1.1	ノードIDの設定	39
5.1.2	ボーレートの設定	40
5.2	オブジェクトディクショナリによるノード番号の設定	40
6	パラメータの説明	41
6.1	CiA 301の通信オブジェクト	41
6.2	メーカー固有のオブジェクト	50

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CANopen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、CAN-BUSの知識があるソフトウェア開発者およびCAN-BUS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
Attr.	属性
CAN	コントローラエリアネットワーク
CiA	CAN in Automation e.V.
COB ID	通信オブジェクト識別子
CS	コマンド指定子
EEPROM	電氣的消去可能プログラマブルROM
EMCY	緊急
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
NMT	ネットワーク管理オブジェクト
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	位置プロファイル
PV	速度プロファイル
ro	リードオンリー
RTR	リモートリクエスト
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
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
SYNC	同期オブジェクト
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

2.1 CANopenデバイスの基本配置

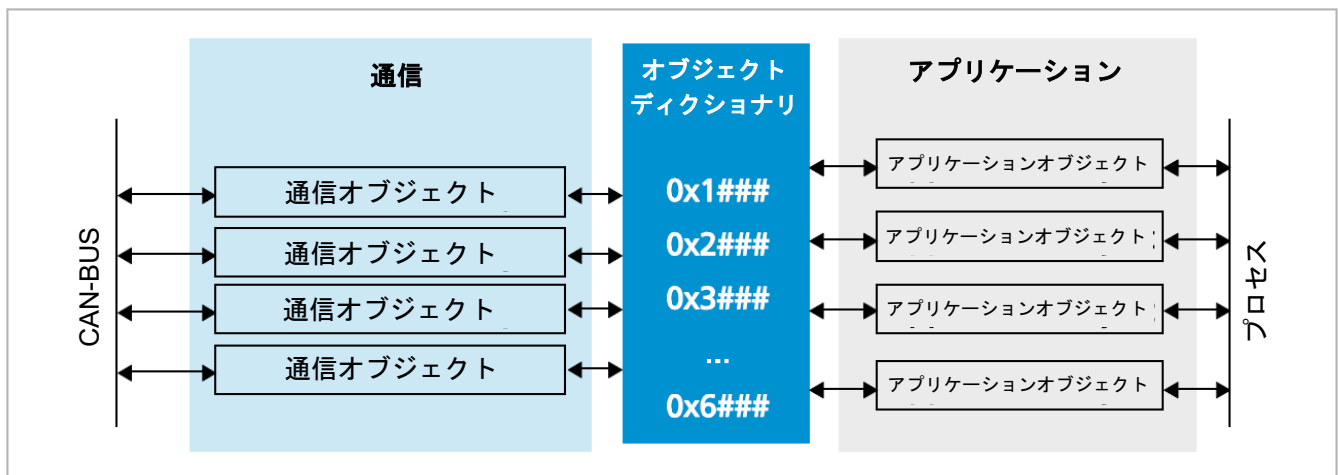


図1: CANopenデバイスの基本配置

通信サービス

CANopenマスターはバスシステムを介し、通信サービスを使用してオブジェクトディクショナリと通信します(3.2節、15ページ参照)。

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（駆動機能）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。

アプリケーション部

アプリケーション部には駆動機能が含まれ、CiA 402に対応しています。駆動機能は、オブジェクト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「駆動機能」のマニュアルを参照してください。

2.2 通信の前提条件

FAULHABER駆動機器は、未設定の状態です。CANネットワークで動作するためには、一意のノード番号を割り当てる必要があります。また、運転時にはボーレートを設定します（5章、39ページ参照）。

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。駆動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（3.8節、27ページ参照）。

1. コントローラを電源に接続します（最低限の電子部品への電源供給）。
2. CAN_H、CAN_L、GNDをホスト側CAN接続の各接続端子に接続します。
3. 電源を投入し、設定アプリケーションから接続を確立します。

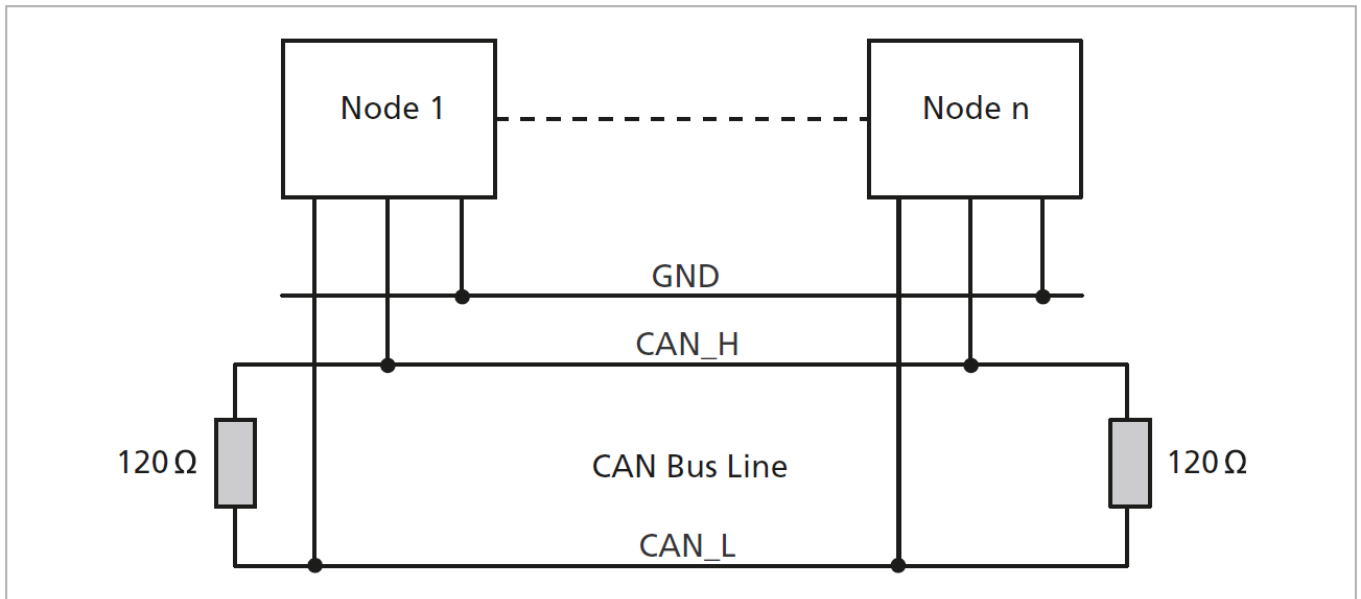


図2: CANopenネットワーク接続

2.3 FAULHABER Motion Manager

FAULHABER駆動機器の最初の運転時は、「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。

FAULHABER Motion Managerを使用すると、接続されたモータとコントローラの設定やパラメータへ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。

ウィザード機能は、駆動コントローラの設定時にユーザをサポートします。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、駆動コントローラに接続されたモータの設定をします。
- コントロール設定ウィザード: 制御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償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。

2.4 パラメータの保存と復元

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（6.1節、41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。

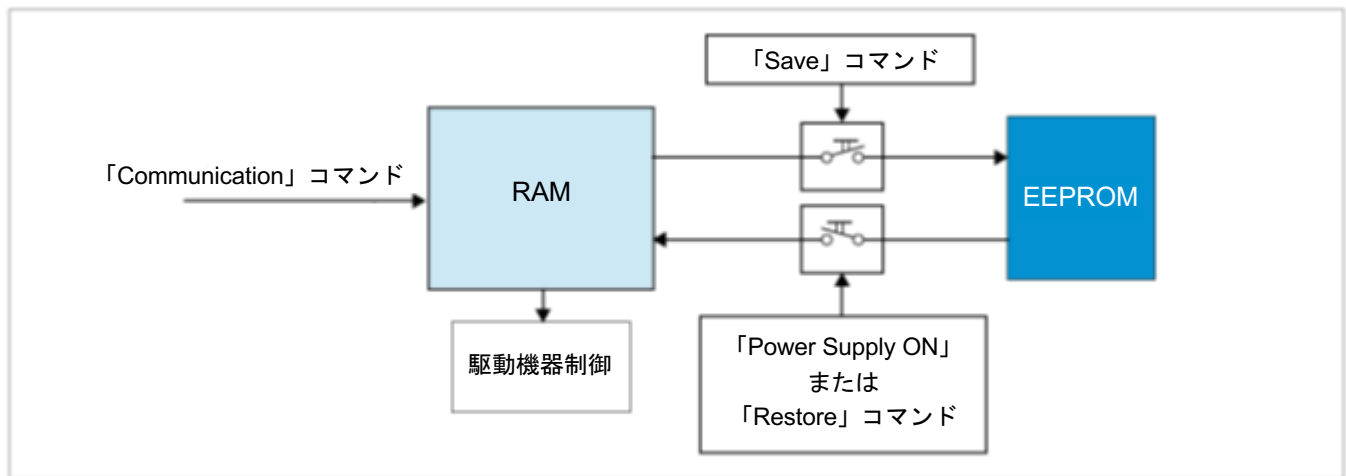


図3: パラメータの保存および復元

「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（6.1章、41ページ参照）。

- 初期設定
- 「Save」コマンドを使用して保存したパラメータ

2.4.1 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21参照）。

- ▶ 「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22参照）。

2.4.2 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23参照）。

- オブジェクト0x1011のサブインデックス01~06に「Load」を書き込みます（表24参照）。
 - 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された値に上書きします。

i 現在ロードしている値を「Restore」後に使用できる可能性がある場合は、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。

2.4.3 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、制御パラメータ、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、頻り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	U32	rw	コントローラのゲイン[1e ⁻⁶]
	0x02	Integral time TN	U16	rw	コントローラリセット時間[100 μs]
0x2346	0x01	Setpoint velocity filter time T _F	U16	rw	フィルタ時間T _F [100 μs]
	0x02	Setpoint filter enable	U8	rw	0: 非アクティブ状態 1: アクティブ状態
0x2347	0x01	Gain factor	U8	rw	ゲイン係数（PPモードでの速度コントローラではK _P を使用） 0: 速度コントローラのゲイン係数は目標値で0になる 128: 可変ゲインなし 255: 速度コントローラのゲイン係数は目標値で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／推力の制御係数 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／推力の制御係数 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	速度プロファイル[ユーザ定義のスケールリング]
0x6083	0x00	Profile acceleration	U32	rw	加速プロファイル[1/s ²]
0x6084	0x00	Profile deceleration	U32	rw	減速プロファイル[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	U16	rw	上限値[相対的なスケールリング]
0x60E1	0x00	Negative torque limit value	U16	rw	下限値[相対的なスケールリング]

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」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」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」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」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。

3 CANopenプロトコルの説明

3.1 はじめに

CANopen

CANopenは標準ソフトウェアプロトコルです。CANopenを使用する通信には、CANハードウェア環境が必要です。CANopenネットワークでは最大127のノードをアドレス指定できます。最大送信速度は1MBit/sです。

CAN標準化

CiA301では、以下の状態がCiAにより定義されています。

- 通信構造
- 制御および監視機能

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

CANopenテレグラムの構造

CANopenテレグラムには11ビットの識別子があり、最大8バイトのユーザデータを格納できます。

表1: CANopenテレグラム構造の概略

11ビットの識別子	最大8バイトのユーザデータ							
11-bit	8-bit	8-bit	8-bit	8-bit	8-bit	8-bit	8-bit	8-bit

3.2 通信サービス

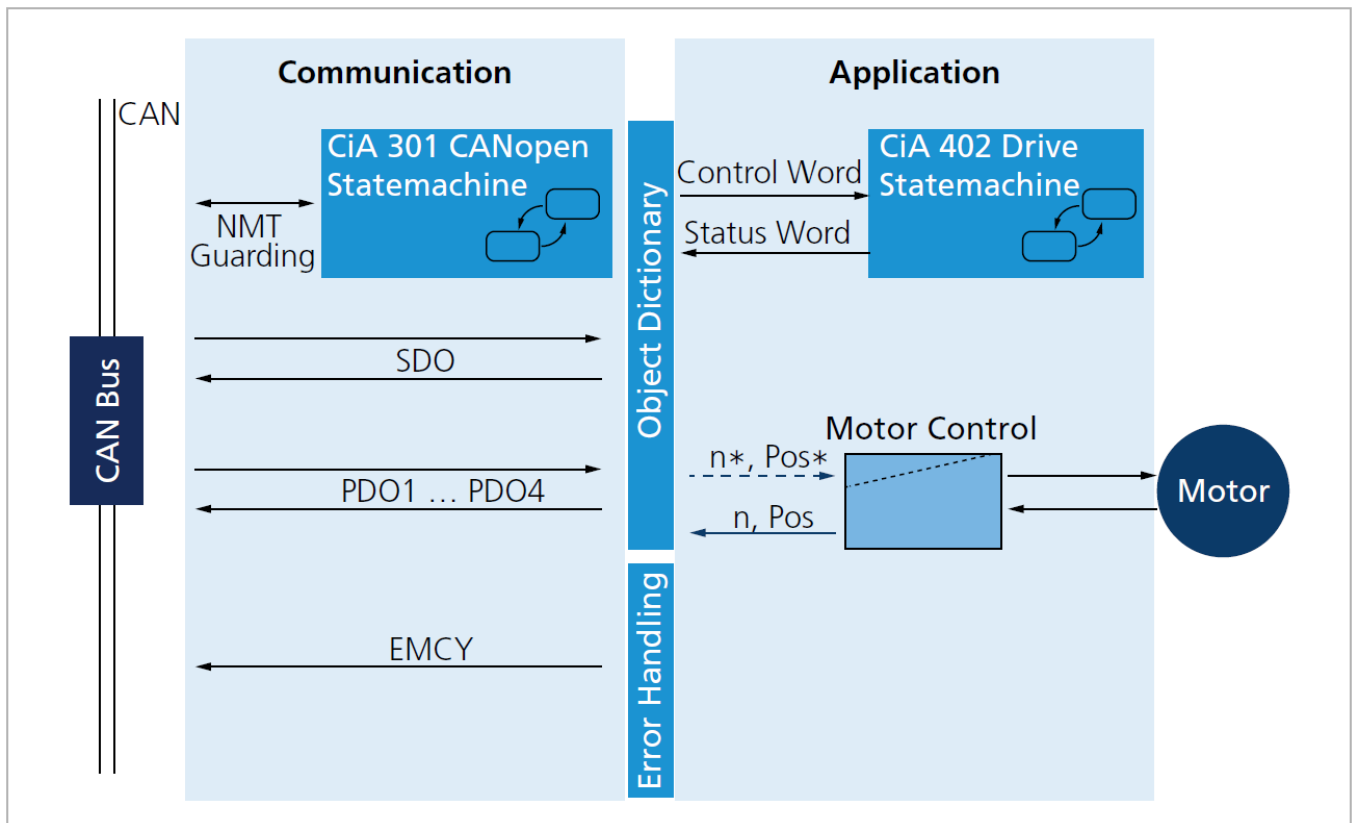


図4: モーションコントローラの通信サービス

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。

表2: CiA 301の通信サービス

通信サービス	詳細
NMT (Network Management)	ノードをアクティブ化し、ノードの現在の状態を監視します (3.8節、27ページ参照)。
SDO (Service Data Object)	CANopenマスターは、SDOを使用してノード内のパラメータにアクセスします。各SDOアクセスでは、1パラメータのみを正確に読み込みまたは書き込みします。SDOはネットワーク内の1つのノードのみをアドレス指定できます (3.5節、21ページ参照)。
PDO (Process Data Object)	PDOはリアルタイムデータへのアクセスに使用され、CANメッセージを使用して同時に複数の駆動機器のパラメータにアクセスできます。PDOで送受信したパラメータは、任意に設定できます (3.4節、17ページ参照)。
SYNC object	SYNCオブジェクトは、CAN-BUSの異なるアプリケーションを同期するために使用されません (3.7節、26ページ参照)。
EMCY (Emergency Object)	緊急メッセージはCANopenマスターにエラーを通知するために使用されます。CANメッセージは、非同期的にエラーコードを伝達し、エラー後にCANopenスレーブの状態の問い合わせが不要になります (3.6節、23ページ参照)。

通信プロファイル

FAULHABERモーションコントローラは、CiA 301 V4に準拠したCANopen通信プロファイルをサポートしています。

- 4個の送信PDO
- 4個の受信PDO
- 1個のサーバSDO
- 緊急オブジェクト

- ノードガードおよびハートビート機能があるNMT
- SYNCオブジェクト

i PDOのデータ割り当ては、CiA 402 V3で規定する「PDO set for servo drive (サーボドライブ用PDOセット)」に事前設定されていますが、ユーザは変更することができます (動的PDOマッピング)。

3.3 識別子の配布

通信オブジェクト識別子 (COB-ID) には7ビットのノードアドレス (Node-ID) と4ビットの機能コードが含まれます。

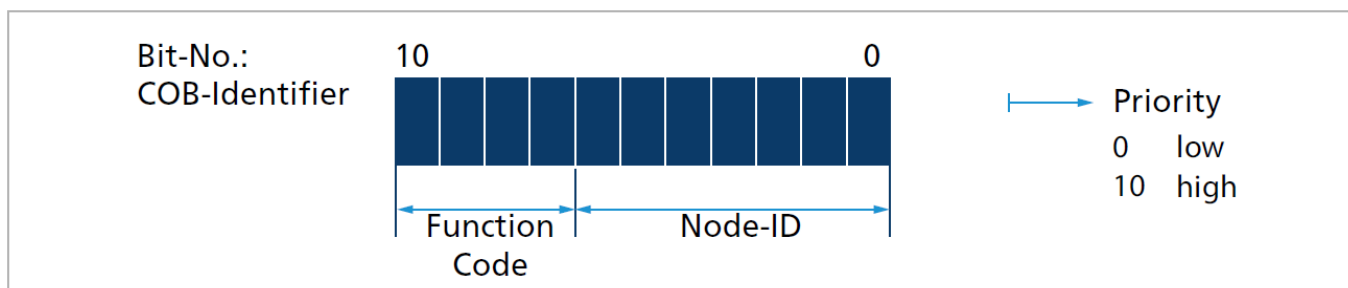


図5: 識別子の配布

定義済み接続設定は、最重要オブジェクトの標準識別子を定義します。

表3: 標準識別子

オブジェクト	機能コード (バイナリ)	結果のCOB-ID	通信設定の オブジェクトインデックス
NMT	0000	0	-
SYNC	0001	128 (80 h)	1005 h
EMERGENCY	0001	129 (81h) to 255 (FFh)	1014 h
PDO1 (tx)	0011	385 (181h) to 511 (1FFh)	1800 h
PDO1 (rx)	0100	513 (201h) to 639 (27Fh)	1400 h
PDO2 (tx)	0101	641 (281h) to 767 (2FFh)	1801 h
PDO2 (rx)	0110	769 (301h) to 895 (37Fh)	1401 h
PDO3 (tx)	0111	897 (381h) to 1023 (3FFh)	1802 h
PDO3 (rx)	1000	1025 (401h) to 1151 (47Fh)	1402 h
PDO4 (tx)	1001	1153 (481h) to 1279 (4FFh)	1803 h
PDO4 (rx)	1010	1281 (501h) to 1407 (57Fh)	1403 h
SDO (tx)	1011	1409 (581h) to 1535 (5FFh)	1200 h
SDO (rx)	1100	1537 (601h) to 1663 (67Fh)	1200 h
NMT error control	1110	1793 (701h) to 1919 (77Fh)	-

PDOのCOB-ID、SYNCオブジェクトおよび緊急オブジェクトは、オブジェクトディクショナリの通信パラメータを使用して変更できますが、SDOテレグラムのCOB-IDは変更できません。IDは常にノード番号から抽出されます。



納入時のシステムはノード番号が1に設定され、この番号に基づきCOB-IDが事前設定されています。

- RxPDO: 201h、301h、401h、501h
- TxPDO: 181h、281h、381h、481h
- EMCY: 81 h
- RxSDO: 581 h
- TxSDO: 601 h



LSSプロトコルを使用してノード番号を変更した場合、PDOのCOB-IDおよび緊急オブジェクトは変更されません。

3.4 PDO（プロセスデータオブジェクト）

PDOは、最大8バイトのユーザデータのCANメッセージです。PDOには、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節、27ページ参照）。

PDOの送信は様々な方法でトリガされます。送信の動作は、オブジェクトディクショナリ内の通信パラメータの送信の種類のパラメータを使用して各PDOに設定できます。

表4: PDO送信の種類

送信の種類	詳細
Event-driven	イベント駆動型RxPDOは受信直後に処理されます。 イベント駆動型TxPDOは、機器の状態ワードが含まれ、変更されたときに送信されます。
Remote request (RTR)	データは、要求メッセージに回答して送信されます。
Synchronised	データはSYNCオブジェクト受信後に送信されます（3.7節、26ページ参照）。

3.4.1 PDOの構造

- 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を使用していない場合）が必要です。
- 送信の種類とPDOのCOB-IDは、オブジェクト0x1400～0x1403および0x1800～0x1803を使用して変更できません。
- 送信の種類のパラメータを使用して、PDOの送信の動作を変更できます。

表5: PDOの送信の種類

送信の種類	意味
255	非同期（イベント駆動型）
253	非同期、要求に応じて（RTR）
1 to 240	同期、周期的 PDOは各SYNCオブジェクトの後に送信されます（3.7節、26ページ参照）。値がSYNCオブジェクトの数と同じであるため、PDOを再送信する前に受信する必要があります（1 = PDOは、各SYNCオブジェクトにつき1個のPDOを送信）。
0	同期、非周期的 PDOの内容が変更されたときは、SYNCオブジェクトの直後にPDOが送信または実行されます（3.7節、26ページ参照）。

3.4.2 標準構成のPDOマッピング（初期設定の状態）

RxPDO1: 制御ワード

11ビットの識別子	2バイトのユーザデータ					
0x200 (512d) + node ID	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、駆動機能のマニュアルを参照してください。

TxPDO1: 状態ワード

11ビットの識別子	2バイトのユーザデータ					
0x180 (384d) + node ID	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、駆動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）

11ビットの識別子	6バイトのユーザデータ					
0x300 (768d) + node ID	LB	HB	LLB	LHB	HLB	HHB

RxPDO2には、16ビットの制御ワードおよび位置プロファイルモード（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

11ビットの識別子	6バイトのユーザデータ					
0x280 (640d) + node ID	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)

11ビットの識別子	6バイトのユーザデータ					
0x400 (1024d) + node ID	LB	HB	LLB	LHB	HLB	HHB

RxPDO3には、16ビットの制御ワードおよび速度プロファイルモード (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

11ビットの識別子	6バイトのユーザデータ					
0x380 (896d) + node ID	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

11ビットの識別子	6バイトのユーザデータ					
0x400 (1024d) + node ID	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

11ビットの識別子	6バイトのユーザデータ					
0x380 (896d) + node ID	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。

3.4.3 マッピングエラーの取り扱い

CiA 301に定義されたマッピング手順に準拠していない場合は、以下のSDOエラーが返されます。

表6: 間違ったマッピング手順によるSDOエラー

SDOエラー	意味	原因
0x06090030	全般的な値範囲エラー	マッピング手順に指定された範囲外のマッピングパラメータ。
0x06020000	オブジェクトディクショナリにオブジェクトがない	マッピングされたオブジェクトの方が、マッピングパラメータオブジェクトの各サブインデックスで有効なエントリ数より大きな値です。

マッピングされたオブジェクトの数が0の場合は、PDOの内部で無効のフラグが付けられ、動作しなくなります。



SDOエラー表に、その他のマッピングエラーが記載されています (3.5.2項、22ページ参照)。

3.4.4 ダミーマッピング

ネットワークに接続された2台以上の機器が、応答できるようにRxPDOを設定できます。この場合、複数の機器のうち1台が、PDOに含まれるデータの一部のみを評価できるのが理想的な形です。

ローカルで使用されていないデータの場合、使用されているデータ型のうちの1つに作成したダミーマッピングをPDOマッピング表に入力できます。

インデックス	型
0x0002	S8
0x0003	S16
0x0004	S32
0x0005	U8
0x0006	U16
0x0007	U32

例

RxPDOには二軸のターゲット位置が含まれます。

一番目のターゲット位置に対応するノードのマッピング:

- 0x160x.00 = 2
- 0x160x.01 = 0x607A0020
- 0x160x.02 = 0x00040020

二番目のターゲット位置に対応するノードのマッピング:

- 0x160x.00 = 2
- 0x160x.01 = 0x00040020
- 0x160x.02 = 0x607A0020

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。クライアント（PC、PLC（プログラマブルロジックコントローラ））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、クライアントからのデータを受信します（ダウンロード）。

表7: SDOユーザ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8: 送信のSDOタイプの分散

送信の種類	バイト数	目的
優先転送	maximum 4 bytes	個別数値パラメータの読み込み／書き込み
セグメント化による転送	more than 4 bytes	テキストパラメータの読み込み（機器名、ファームウェアバージョンなど）とデータブロックの送信（追跡バッファなど）

本マニュアルでは優先転送のみを説明します。セグメント化による転送については、CiA 301を参照してください。

3.5.1 優先転送

SDOメッセージのサイズは常に8バイトです。

読み込みODエントリ（クライアントからサーバ、アップロード要求）

11ビットの識別子	8バイトのユーザデータ							
0x600 (1536d) + node ID	0x40	Index LB	Index HB	Subindex	0	0	0	0

サーバからクライアント、アップロード応答

11ビットの識別子	8バイトのユーザデータ							
0x580 (1408d) + node ID	CS(0x4x)	Index LB	Index HB	Subindex	LLB (D0)	LHB (D1)	HLB (D2)	HHB (D3)

コマンド指定子CS（0x4x）は、D0~D3の有効なデータバイト数および転送コードを指定します。コマンド指定子は、以下のようにコード化されています。

- CS = 0x4F、D0に1データバイト
- CS = 0x4B、D0~D1に2データバイト
- CS = 0x47、D0~D2に3データバイト
- CS = 0x43、D0~D3に4データバイト

書き込みODエントリ（クライアントからサーバ、ダウンロード要求）

11ビットの識別子	8バイトのユーザデータ							
0x600 (1536d) + node ID	CS(0x2x)	Index LB	Index HB	Subindex	LLB (D0)	LHB (D1)	HLB (D2)	HHB (D3)

コマンド指定子CS（0x2x）は、D0~D3の有効なデータバイト数および転送コードを指定します。コマンド指定子は、以下のようにコード化されています。

- CS = 0x2F、D0に1データバイト
- CS = 0x2B、D0~D1に2データバイト

- CS = 0x27、D0～D2に3データバイト
- CS = 0x23、D0～D3に4データバイト
- CS = 0x22、データバイト数の指定なし

サーバからクライアント、ダウンロード応答

11ビットの識別子	8バイトのユーザデータ							
0x580 (1407d) + node ID	0x60	Index LB	Index HB	Subindex	0	0	0	0

SDOエラー発生時の中断

SDO中断、クライアントからサーバ

11ビットの識別子	8バイトのユーザデータ							
0x600 (1536d) + node ID	0x80	Index LB	Index HB	Subindex	ERROR 0	ERROR 1	ERROR 2	ERROR 3

SDO中断、サーバからクライアント

11ビットの識別子	8バイトのユーザデータ							
0x580 (1536d) + node ID	0x80	Index LB	Index HB	Subindex	ERROR 0	ERROR 1	ERROR 2	ERROR 3

3.5.2 SDOエラーの詳細

ページの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（3.5.1項、21ページ参照）。エラーの種類は、以下のようにコード化されます。

- Error0: 追加エラーコードHB
- Error1: 追加エラーコードLB
- Error2: エラーコード
- Error3: 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	PDO長以上のマッピングされたオブジェクトの数および/または長さ
0x06	0x04	0x0043	パラメータの全般的な非互換性
0x06	0x04	0x0047	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、長過ぎるパラメータ長
0x06	0x07	0x0013	データの種類の不一致、短過ぎるパラメータ長
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

エラー クラス	エラー コード	追加コード	詳細
0x06	0x09	0x0031	全般的な範囲エラー: 大き過ぎるパラメータ値
0x06	0x09	0x0032	全般的な範囲エラー: 小さ過ぎるパラメータ値
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）

緊急オブジェクトは、問い合わせを必要とせず、他のバスに構成された機器に対して非同期的にエラーを通知します。緊急オブジェクトは常に8バイトのサイズです。

11ビットの識別子	8バイトのユーザデータ							
0x80 (128d) + node ID	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、6.1節、41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12参照）
- 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、各エラーコードに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。

表9に、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。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ます（表13参照）。

表9: 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	No error (is sent out when an error is no longer present or has been acknowledged)	-	-	-	-	-
-	-	-	-	-	0	Generic error (is set if one of the error bits 1 to 7 is set)
0x3210	Overvoltage	0x0004	2	Overvoltage error	2	Voltage error
0x3220	Undervoltage	0x0008	3	Undervoltage error	2	Voltage error
0x43F0	Temperature warning	0x0010	4	Temperature warning	1	Current error ^{a)}
0x4310	Temperature error	0x0020	5	Temperature error	3	Temperature error
0x5410	Output stages	0x0080	7	IntHW error	7	Manufacturer-specific error
0x5530	EEPROM fault	0x0400	10	Memory error	-	-
0x6100	Software error	0x1000	12	Calculation error	7	Manufacturer-specific error
0x7200	Measurement circuit: Current measurement	0x0200	9	Current measurement error	7	Manufacturer-specific error
0x7300	Sensor fault (encoder)	0x0040	6	Encoder error	7	Manufacturer-specific error
0x7400	Computation circuit: Module fault	0x0100	8	Module error	7	Manufacturer-specific error
0x8110	CAN overrun	0x0800	11	Communications error	4	Communications error
0x8130	CAN guarding failed					
0x8140	CAN recovered from bus stop					
0x8310	RS232 overrun					
0x84F0	Deviation error (velocity controller)	0x0001	0	Speed deviation error	5	Drive-specific error
0x84FF	Max Speed Error	0x2000	13	DynamicError	7	Manufacturer-specific error
0x8611	Following error (position controller)	0x0002	1	Following error	5	Drive-specific error

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10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）の下にセットされま
す（表14参照）。
- 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（駆動機能のマニュアルを参照してくだ
さい）。

表10: 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 SYNCオブジェクト

SYNCオブジェクトはユーザデータのないメッセージです。SYNCオブジェクトは、同期したPDOをトリガします。それと同時にさまざまな機器のプロセスを開始します。

SYNCオブジェクトの識別子が、オブジェクトディクショナリのインデックス0x1005にセットされます（初期設定0x80）。

11ビットの識別子	0バイトのユーザデータ
0x80	no user data

i SYNCオブジェクトがPDOをトリガするには、PDOの送信の種類を適切に設定する必要があります（表5参照）。

3.7.1 同期PDOのトリガ

同期RxPDO: PDOとともに送信されたコマンドは、SYNCオブジェクトが受信されるまで実行されません。RxPDOの送信の種類1~240は、送信の種類0と同一です。

同期TxPDO: 現在のデータとともに送信されたPDOは、SYNCオブジェクトが受信されるまで実行されません。

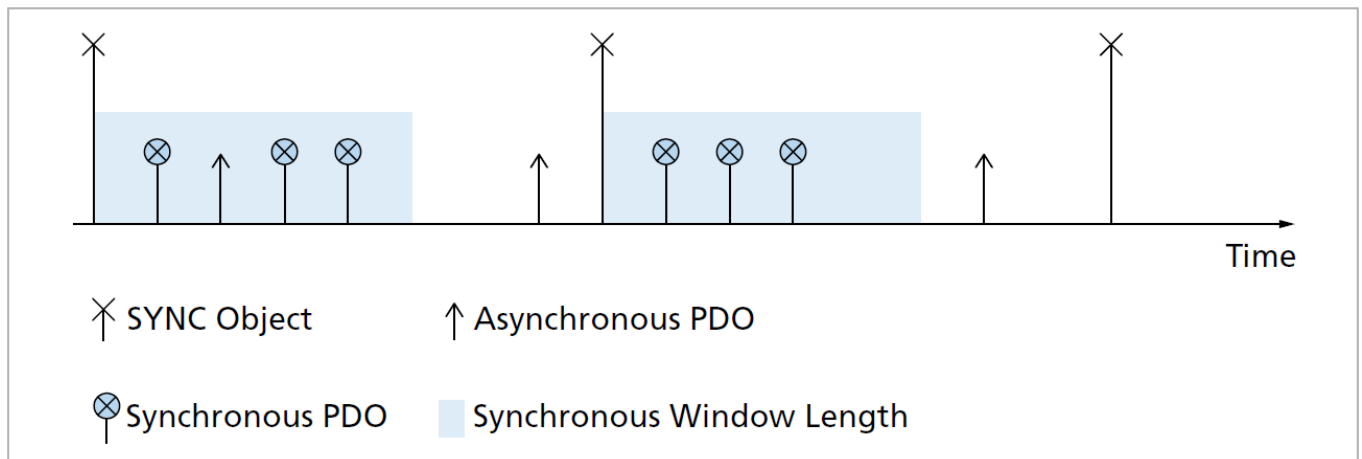


図6: ウィンドウの長さのチャート

i ノードは送信の種類1~240でグループ化することもできます。

モーションコントローラは、受信PDOの同期したウィンドウの長さをチェックしません。SYNCオブジェクトの受信直後に利用可能なTxPDOを送信します。

3.8 NMT (ネットワーク管理)

ネットワーク管理オブジェクトは、CANopen機器のCiA 301機器状態を管理し、ネットワークノードを監視します。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 (Pre-Operational)** 状態になります。**運転前**状態では、機器はSDOを介してNMTメッセージのみと通信できます。

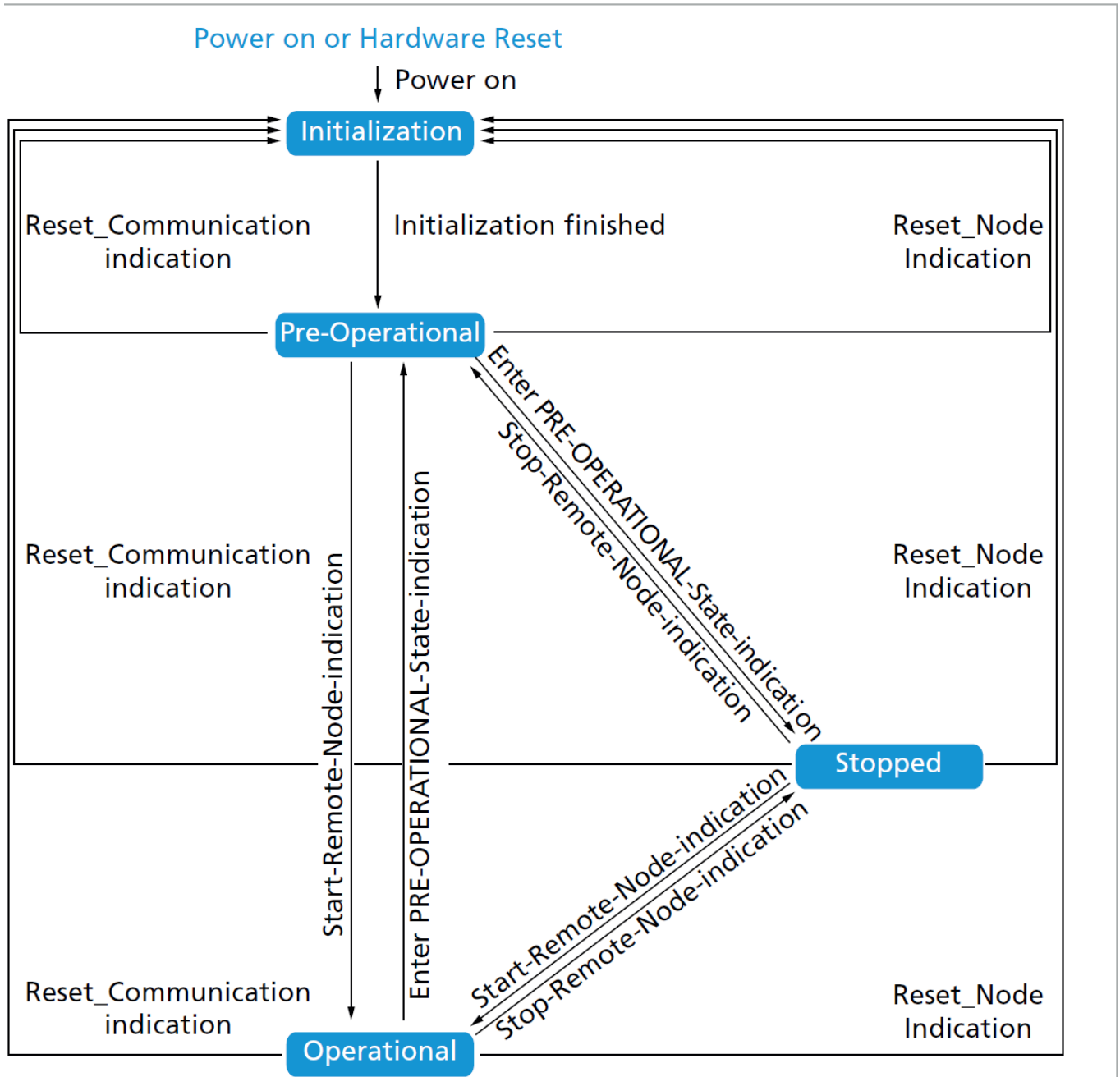


図7: CiA 301機器状態

表11: NMTの状態変更

状態の遷移	CS	意味
Power on	-	電源投入時に自動で初期化状態になります。
Initialisation finished	-	初期化後に機器は自動的に運転前状態になり、起動メッセージを送信します。
Start remote node indication	0x01 (1d)	機器が起動し、PDO送信が可能になります。
Enter Pre-operational state indication	0x80 (128d)	PDO送信を停止しますが、SDOはアクティブな状態です。
Stop remote node indication	0x02 (2d)	駆動機器が停止状態になり、SDOおよびPDOはオフに切り替わります。
Reset node indication	0x81 (129d)	リセットを実行します。全てのオブジェクトが標準のスイッチオンにリセットされます。
Reset communication indication	0x82 (130d)	通信機能のリセットを実行します。

i FAULHABERモーションコントローラは、全てのオブジェクトの標準構成を備えています。初期設定が完了すると、アプリケーション固有の設定を機器に直接保存できます。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。

CANopenノードの開始

リモートノードの開始:

11ビットの識別子	2バイトのユーザデータ						
0x000	0x01	Node ID					

CANメッセージでネットワーク全体を開始させることもできます。

全てのリモートノードの開始:

11ビットの識別子	2バイトのユーザデータ						
0x000	0x01	0x00					

ネットワーク全体またはノードを開始させると、機器は**運転状態**になります。PDOを使用して機器を動作させることができます。

停止状態の場合、機器はエラー状態であるためPDOを使用して動作できません。この場合、機器との通信はNMTメッセージのみとなります。

NMTメッセージは、常に識別子0x000の2バイトで構成されます。

NMTメッセージ

11ビットの識別子	2バイトのユーザデータ						
0x000	CS	Node ID					

ユーザデータの割り当て:

- CSコマンド指定子 (表11参照)
- ノードID: ノードアドレス (0 = 全てのノード)


i 重大な通信エラーが発生した場合、モーションコントローラは設定で**運転前**NMT状態に切り替わります。オブジェクト0x1029を使用して異なる動作を設定することもできます。

3.8.1 起動

開始フェーズの直後に、モーションコントローラにより起動メッセージが送信されます。起動メッセージは、モジュールの電源投入後に開始フェーズの終了を通知します。起動メッセージは、ノードガードのメッセージの識別子が1データバイト（バイト0 = 0x00）のCANメッセージです（0x700 + ノードID）。

11ビットの識別子	1バイトのユーザデータ						
0x700 (1792d) + node ID	0x00						

3.8.2 監視機能

 一度に使用できる監視機能は、ノードガードまたはハートビートのいずれか一方です。

3.8.2.1 ノードガード

ノードガードのオブジェクトは、機器の瞬間的状态を問い合わせます。問い合わせをするには、マスターが監視対象のノードのガード識別子を要求するとともに、リモートフレームを設定します。監視対象のノードが、ノードの現在の状況およびトグルビットを含むガードメッセージで応答します。

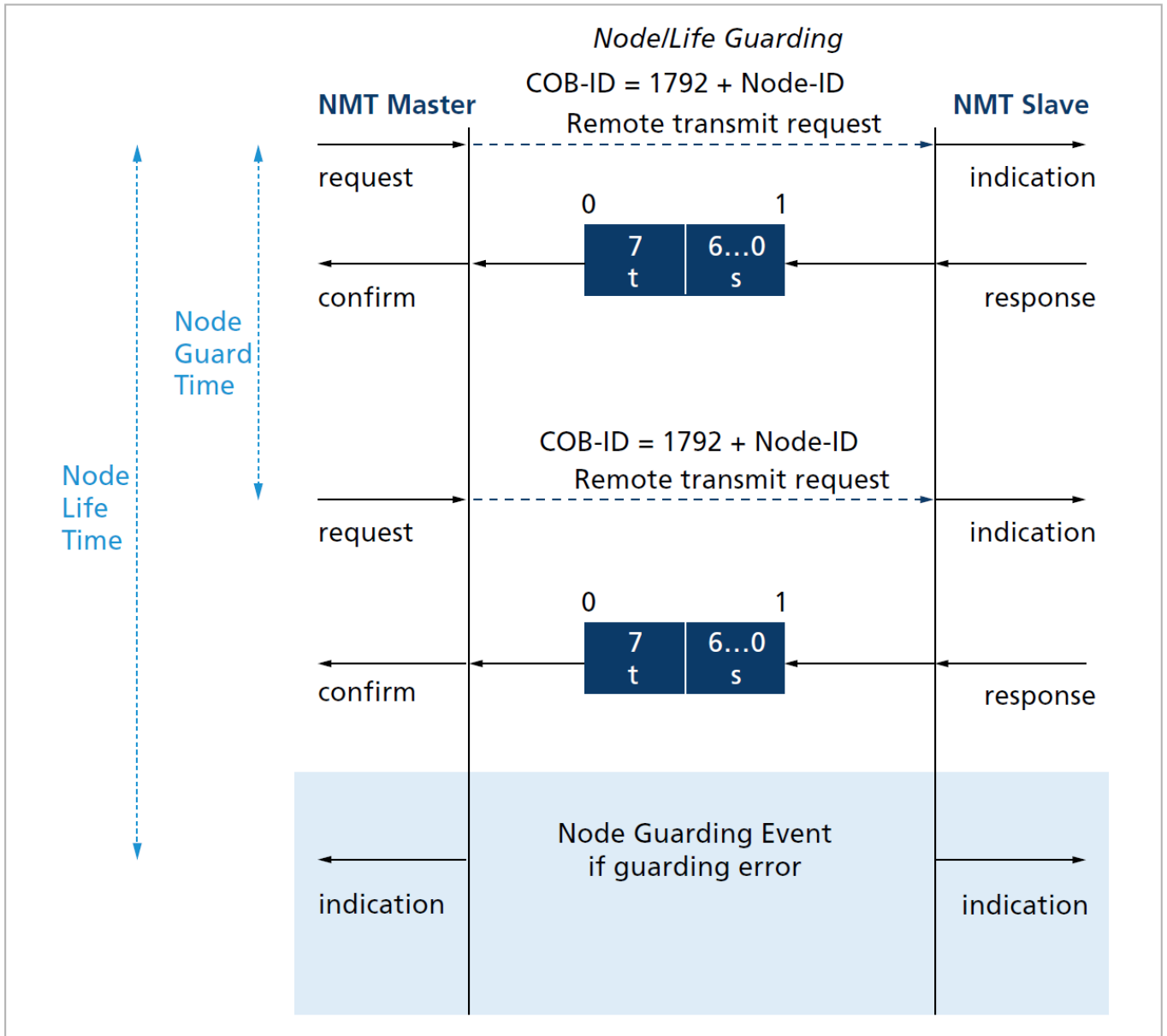


図8: ノードガードプロトコルの図

t: トグルビット

初期値は0で、各ガードテレグラムで値が変化します

s: 状態

s = 0x04 (4d): 停止

s = 0x05 (5d): 動作

s = 0x7F (127d): 運転前

ノード寿命 > 0 を設定 (オブジェクト0x100Cおよび0x100D) し、設定した寿命期間中にマスターがノードガードを要求しなかった場合、ノードガードのエラーがセットされます。ノードガードのエラーへの応答が、FAULHABERエラーレジスタ (オブジェクト0x2321) によりセットされます (表14参照)。初期設定は、緊急メッセージ0x8130の送信です。

3.8.2.2 ハートビート

モーションコントローラがハートビートプロデューサとハートビートコンシューマの両方として動作するように設定できます。

- **ハートビートプロデューサ:** モーションコントローラが周期的にメッセージを送信し、メッセージはネットワークのハートビートコンシューマによって受信されます。
- **ハートビートコンシューマ:** ハートビートコンシューマ時間内に監視対象のハートビートプロデューサからハートビートメッセージを受信しなかった場合、モーションコントローラはFAULHABERエラーレジスタ（オブジェクト0x2320）で指定される動作で応答します（表12参照）。

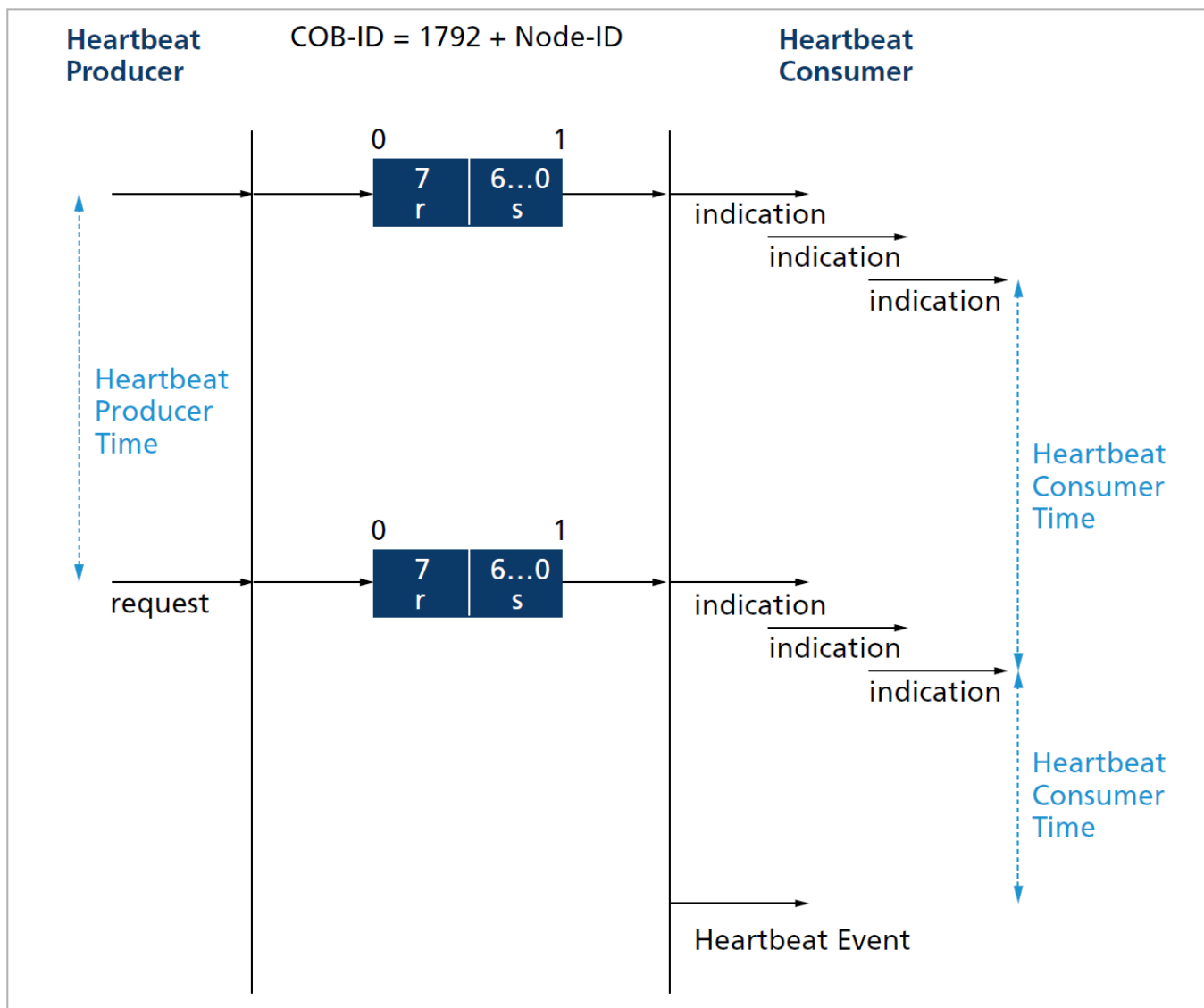


図9: ハートビートプロトコルのチャート

r: 予約済み

常に0

s: 状態

s = 0x00 (0d): 起動

s = 0x04 (4d): 停止

s = 0x05 (5d): 動作

s = 0x7F (127d): 運転前

3.8.3 監視機能の設定

- 一度に1つの監視機能（ノードガードまたはハートビート）のみアクティブにできます。
- ハートビートプロデューサ時間 > 0（オブジェクト0x1017）の場合、モーションコントローラはハートビートプロデューサとして動作します。モーションコントローラは、ハートビートプロデューサ時間の間隔でハートビートメッセージを送信します。ノードガードの時間は0に設定されます（3.8.2.1項、29ページ参照）。
- ハートビートをアクティブ化した場合、電源投入後の起動メッセージは最初のハートビートメッセージとしてランク付けされます。ハートビートは、ハートビートプロデューサ時間の間隔で順次送信されます。
- ハートビートプロデューサ時間に加えて、ハートビートコンシューマ時間 > 0（オブジェクト0x1016.01）が設定されている場合、モーションコントローラはハートビートコンシューマとして動作します。ハートビートプロデューサの設定は無効となります。監視対象のマスターのノードIDおよびハートビートコンシューマ時間が、オブジェクト0x1016に入力されます。
- ハートビートコンシューマ時間は常に、マスターのハートビートプロデューサ時間より長く設定する必要があります。
- モーションコントローラが、設定したハートビートコンシューマ時間以内にマスターからハートビートメッセージを受信しない場合は、ハートビートイベントがトリガされます。ハートビートイベントへの応答が、FAULHABERエラーレジスタのエラーマスク（オブジェクト0x2321）により決定されます（表12参照）。初期設定は、緊急メッセージ0x8130の送信です。
- ハートビートプロデューサがアクティブ化されている間に、ノードガードの時間の設定をしようとした場合、SDOエラー0x08000020（アクセス不可）が送信されます。

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により参照できます（SDOプロトコル）。

- 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（6.1節、41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（6.2節、50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（駆動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 CANエラー

CANオーバーラン（オブジェクト喪失）

メッセージを喪失した場合、コントローラは緊急メッセージ0x8110を送信します。ビット4（通信エラー）がエラーレジスタにセットされ、ビット7（CANオーバーラン）がFAULHABERエラーレジスタにセットされます。緊急メッセージは遅延の後に送信されます。緊急メッセージ（0x000）の発行によって、エラーが取り消されることはありません。エラーレジスタおよびFAULHABERエラーレジスタの各ビットは消去されません。

エラー受動モードのCAN

駆動機器のCANモジュールが**エラーパッシブ**状態に設定されると、緊急メッセージ0x8120が送信されます。ビット4（通信エラー）がエラーレジスタにセットされ、ビット6（エラーパッシブモードのCAN）がFAULHABERエラーレジスタにセットされます。緊急メッセージ（0x000）が送信され、駆動機器が**エラーアクティブ**状態に復元されるとエラーが取り消されます。

バス停止からの復旧

駆動機器のCANモジュールが、**バスオフ**状態中に有効なメッセージを受信すると、緊急メッセージ0x8140が送信され、**バスオフ**状態の終了を報告します。ビット4（通信エラー）がエラーレジスタにセットされ、ビット9（バスオフからの復旧）がFAULHABERエラーレジスタにセットされます。これによりエラーは取り消されません。エラーレジスタおよびFAULHABERエラーレジスタの各ビットは消去されません。

i 「CANオーバーラン」および「バスオフからの復旧」は、重大な通信エラーです。エラーレジスタおよびFAULHABERエラーレジスタの各ビットは、モーションコントローラの再起動でのみ消去できます。その他の重大な通信エラーを以下に示します。

- ノードガードのタイムアウト
- ハートビートのタイムアウト

3.10.2 機器不良

表12: 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13: 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	低電圧検出
0x0010	Temperature warning	警告が出力される温度を超えた
0x0020	Temperature error	エラーメッセージが出力される温度を超えた
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

エラービット	エラーメッセージ	詳細
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	設定最大速度より現在のモータ速度が速い
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、23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13参照)。

表14: 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	駆動機器の機器状態を <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	駆動機器の機器状態を <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、過電流がエラー状態に設定されるため、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。

4 追跡

追跡機能を使用すると、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。2種類の記録方法が利用できます。

- 追跡レコーダ: 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 (4.1節、35ページ参照)。
- 追跡ロガー: 要求に応じてパラメータ値が要求され、連続して読み込まれます (4.2節、38ページ参照)。



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。

4.1 追跡レコーダ

追跡レコーダによるデータの設定および読み込みはSDOを使用して実行されます。

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。

記録されたデータは、セグメント化SDOアップロードプロトコルを使用して読み込まれます。ODオブジェクト0x2371はこの目的のためにODで利用可能です。(4.1.2項、37ページ参照)。

4.1.1 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。ここでは、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15: 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	記録するサンプリングレート 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) 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。対応するオブジェクトエントリのポインタとして、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 (必要なパラメータのインデックスとサブインデックス) に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 (実位置値) 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: 記録は、セットされたサンプリング時間の倍数で定義された時間に開始されます。
- 遅延 < 0: バッファの長さまで負の遅延が実行されます。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点で終了します。これにより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16: 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がオブジェクト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。データソースごとに最大2kBのバッファが可能です。

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。


4.1.2 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17: 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 (ODエントリによる) およびセットされたバッファサイズのデータ長により依存するため、記録された値を読み込むには、各データソースのデータ長のサイズにバッファサイズを掛けたメモリーエリアが必要になります。

 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。

トリガ状態 (0x2371.01)

表18: トリガ状態 (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、トリガ状態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 (状態 = 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。

4.1.3 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. 必要に応じて、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. トリガの立上りもしくは立下りを設定し、記録 (2370.04) をアクティブ化します。
↳ 以上で追跡レコーダの設定は完了です。
7. 値3のトリガ状態 (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。

4.2 追跡ロガー

追跡ロガーは、PDO通信サービスを使用してコントローラからデータを送信します。任意のTxPDOを追跡PDOとして使用できます。使用する前に、PDOマッピング方法を使用してPDOに使用するパラメータをロードする必要があります (3.4.2項、18ページ参照)。その後、リモート要求またはSYNCを使用してデータを周期的に要求できます。

 個別のデータポイントの分解能は、送信および処理速度によって異なります。データポイントの分解能は最小1msです。

5 通信設定

FAULHABER駆動機器は、標準設定でノード番号が1、自動ボーレート検出（AutoBaud）が設定された状態で納入されます。

i ネットワークモードでは、使用中のネットワーク送信レートを固定レートとして設定してください。

5.1 CANネットワーク経路の設定

CANネットワーク経路で設定するには、FAULHABER Motion ManagerまたはCiA 305に準拠したLSSプロトコル（層設定サービスおよびプロトコル）をサポートするその他の設定ツールが必要です。

i FAULHABER Motion ManagerをCANインターフェースを備えたPCにインストールする必要があります。

通信パラメータは、2種類の設定方法があります。

- 各駆動機器は、設定ツールのCANインターフェースで接続されます。
「LSS Switch Mode Global」では、更なるデータがなくても駆動機器が設定モードに切り替わり、ノード番号およびボーレートを設定できます。
- 構成対象の駆動機器は、CANインターフェースを使用してネットワーク経路で設定ツールに接続されます。
「LSS Switch Mode Selective」では、LSSアドレス（ベンダーID、製品コード、改訂番号、シリアル番号）を入力すると、使用する駆動機器のアドレス指定ができます。また、設定モードに切り替えると、ノード番号およびボーレートを設定できます。

MC V3.0までのFAULHABER駆動機器には以下の入力が必要です。

- ベンダーID: 327
- 製品コード: 48
- 改訂番号: 1.0
- シリアル番号: 製品のステッカー参照

LSSプロトコルは、ノード番号およびボーレートの設定以外に、接続されたユニットのLSSアドレスの読み込みおよびノードID設定の読み込みもサポートしています。

LSS通信には、識別子0x7E5（マスター）および識別子0x7E4（スレーブ）を使用します。

構成後、モーションコントローラは設定したパラメータをEEPROMに保存します。このパラメータは、電源の入切後も使用できます。

LSSプロトコルの詳細な説明は、CiA 305を参照してください。

5.1.1 ノードIDの設定

- ノード番号1～127を設定できます。
- ノードID 255（0xFF）は、ノードが構成されていないことを示します。ノードをオンに切り替えると、有効なノード番号が割り当てられるまでノードはLSS初期化状態になります。有効なノード番号が割り当てられると、NMT初期化が実行されます。

5.1.2 ボーレートの設定

- 自動ボーレート検出（AutoBaud）がアクティブな場合、駆動機器は表19に示す通信速度でネットワークに接続します。24個のテレグラム（ボーレート当たり3個）がバスケーブルを通過する前に、ネットワークのボーレートが検出されます。その後駆動機器は、ネットワークのボーレートに一致するように自らを設定します。
- 自動ボーレート検出がアクティブな場合、ボーレートが検出されるまでテレグラムは処理できません。自動ボーレート検出がアクティブな場合、システムの起動時間が長くなります。
- 表19に示す固定ボーレートは、インデックス0~8を入力して設定することができます。

表19: ビットタイミングパラメータ

ボーレート	インデックス
1000 kBit/s	00
800 kBit/s	01
500 kBit/s	02
250 kBit/s	03
125 kBit/s	04
50 kBit/s	06
20 kBit/s	07
10 kBit/s	08
AutoBaud	09

5.2 オブジェクトディクショナリによるノード番号の設定

ノード番号の設定では、CANネットワーク経由のLSS通信以外に駆動機器が利用可能なインターフェース（CAN、USB、RS232）を使用することができます。

オブジェクトディクショナリにオブジェクト0x2400.03を書き込み設定します。

表20: CAN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	4	オブジェクトエントリの数
	0x01	CAN rate	U8	rw	9	表19のCANボーレートのインデックス
	0x03	Node ID U8	U8	rw	255	ノード番号

オブジェクト0x2400.01を使用して、ボーレートの現在の設定（AutoBaudまたは固定ボーレート）を読み込むことができます。

オブジェクト0x2400.03によるノード番号の変更は、最後のノード番号で確認されます。変更されたノード番号はアプリケーションパラメータの「Save」コマンドを実行し、その後「Reset」コマンドを実行するまでロードされません。

6 パラメータの説明

6.1 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情報が含まれています。

- 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、最新のエラーの記録をビットコード形式で格納されています。

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、23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。

COB-ID SYNC

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
0x1005	0x00	COB ID SYNC	U32	rw	0x80	SYNCオブジェクトのCANオブジェクト識別子

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

メーカーのデバイス名を指定するには、セグメント化SDOの記録を読み込む必要があります。

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

メーカーのハードウェアバージョンを決定するには、セグメント化SDOの記録を読み込む必要があります。

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	Manufacture's software version	Vis string	const	-	ソフトウェアバージョン

メーカーのソフトウェアバージョンを決定するには、セグメント化SDOの記録を読み込む必要があります。

ガード時間

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
0x100C	0x00	Guard time	U16	rw	0	ノードガードの監視時間

ガード時間はミリ秒で表示されます。値0はノードガードをオフに切り替えます（3.8.2.1項、29ページ参照）。

寿命係数

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
0x100D	0x00	Life time factor	U8	rw	0	ノードガードの時間係数

寿命係数にガード時間を掛けるとノードガードの寿命時間を求めることができます（3.8節、27ページ参照）。値0は、ノードガードをオフに切り替えます。

パラメータの保存

表21: 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（オブジェクトディクショナリエントリ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（オブジェクトディクショナリエントリ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」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22: 「Save」

signature	ISO 8 859 (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には10,000の書き込みサイクルを収録できます。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ません。

- ▶ 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23: 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)
	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」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24: 「Load」

signature	ISO 8 859 (ASCII)	16進法
MSB	d	64 h
	a	61 h
	o	6Fh
LSB	i	6Ch

 出カステージをオフにすると初期の状態をロードできます。

COB-ID緊急メッセージ

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
0x1014	0x00	COB-ID EMCY	U32	rw	0x80 + node ID	緊急オブジェクトのCANオブジェクト識別子

ハートビートコンシューマ時間

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
0x1016	0x00	Number of entries	U8	ro	1	オブジェクトエントリの数
	0x01	Consumer heartbeat time	U32	rw	0	ハートビート監視時間

- ビット0～15には、ハートビートコンシューマ時間（ミリ秒）が含まれます。値を0にセットすると、ハートビートコンシューマ時間は非アクティブになります（3.8.2.2項、31ページ参照）。
- ビット16～23には、ハートビートメッセージが送信されるノード番号（マスターノードID）が含まれていません。
- ビット24～31は未使用（予備）です。

ハートビートプロデューサ時間

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
0x1017	0x00	Producer heartbeat time	U16	rw	0	ハートビート送信時間の間隔

ハートビートプロデューサ時間のオブジェクトには、ハートビートプロデューサ時間の間隔（ミリ秒）が含まれます。値を0にセットすると、ハートビートプロデューサ時間は非アクティブになります（3.8.3項、32ページ参照）。

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	ベンダー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

エラー時の 挙動

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
0x1029	0x00	Number of entries	U8	ro	1	オブジェクトエントリの数
	0x01	Communication error	U8	rw	0	通信エラー発生時の動作 0 = 運転前状態 1 = 状態変更なし 2 = 停止状態

重大な通信エラーが発生すると、モーションコントローラは、**運転前**NMT状態に切り替わります。サブインデックス1をセットすると、重大な通信エラー発生時の動作を変更できます。

サーバSDO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
0x1200	0x00	Number of entries	U8	ro	2	オブジェクトエントリの数
	0x01	COB ID client to server (rx)	U32	ro	0x600 + node ID	サーバRxSDOのCANオブジェクト識別子
	0x02	COB ID server to client (tx)	U32	ro	0x580 + node ID	サーバTxSDOのCANオブジェクト識別子

PDO1パラメータ受信

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
0x1400	0x00	Number of entries	U8	ro	2	オブジェクトエントリの数
	0x01	COB ID used by RxPDO1	U32	rw	0x200 + node ID	サーバRxPDO1のCANオブジェクト識別子
	0x02	Transmission type	U8	rw	255 (asynchr.)	PDO送信の種類

PDO2パラメータ受信

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
0x1401	0x00	Number of entries	U8	ro	2	オブジェクトエントリの数
	0x01	COB ID used by RxPDO2	U32	rw	0x300 + node ID	サーバRxPDO2のCANオブジェクト識別子
	0x02	Transmission type	U8	rw	255 (asynchr.)	PDO送信の種類

PDO3パラメータ受信

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
0x1402	0x00	Number of entries	U8	ro	2	オブジェクトエントリの数
	0x01	COB ID used by RxPDO3	U32	rw	0x400 + node ID	サーバRxPDO3のCANオブジェクト識別子
	0x02	Transmission type	U8	rw	255 (asynchr.)	PDO送信の種類

PDO4パラメータ受信

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
0x1403	0x00	Number of entries	U8	ro	2	オブジェクトエントリの数
	0x01	COB ID used by RxPDO4	U32	rw	0x500 + node ID	サーバRxPDO4のCANオブジェクト識別子
	0x02	Transmission type	U8	rw	255 (asynchr.)	PDO送信の種類

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パラメータ送信

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
0x1800	0x00	Number of entries	U8	ro	2	オブジェクトエントリの数
	0x01	COB ID used by TxPDO1	U32	rw	0x180 + node ID	TxPDO1のCANオブジェクト識別子
	0x02	Transmission type	U8	rw	255 (asynchr.)	PDO送信の種類

PDO2パラメータ送信

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
0x1801	0x00	Number of entries	U8	ro	2	オブジェクトエントリの数
	0x01	COB ID used by TxPDO2	U32	rw	0x280 + node ID	TxPDO2のCANオブジェクト識別子
	0x02	Transmission type	U8	rw	255 (asynchr.)	PDO送信の種類

PDO3パラメータ送信

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
0x1802	0x00	Number of entries	U8	ro	2	オブジェクトエントリの数
	0x01	COB ID used by TxPDO3	U32	rw	0x380 + node ID	TxPDO3のCANオブジェクト識別子
	0x02	Transmission type	U8	rw	255 (asynchr.)	PDO送信の種類

PDO4パラメータ送信

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
0x1803	0x00	Number of entries	U8	ro	2	オブジェクトエントリの数
	0x01	COB ID used by TxPDO4	U32	rw	0x480 + node ID	TxPDO4のCANオブジェクト識別子
	0x02	Transmission type	U8	rw	255 (asynchr.)	PDO送信の種類

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	

6.2 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

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。

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	駆動機器の機器状態を <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	駆動機器の機器状態を <i>Quick Stop Active</i> 状態に切り替えるエラー

駆動機器の機器状態は、駆動機能のマニュアルで説明されています。

追跡構成

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	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

CANボーレートインデックスおよびノード番号

表25: CAN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	4	オブジェクトエントリの数
	0x01	CAN rate	U8	rw	9	表19のCANボーレートのインデックス
	0x03	Node ID	U8	rw	255	ノード番号

〒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

7000.05050 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