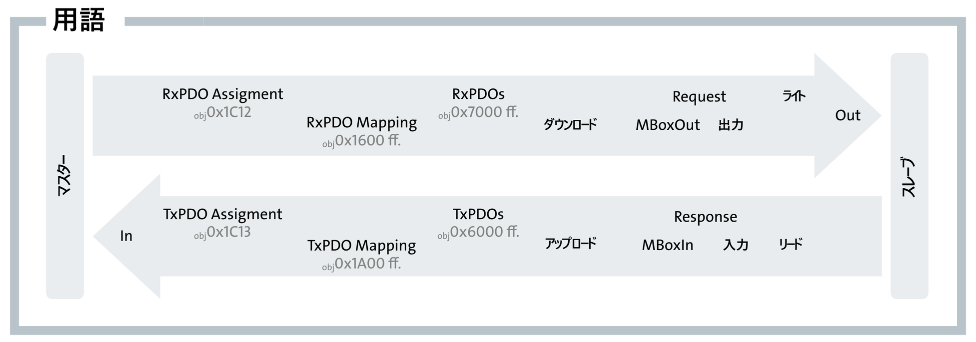
EtherCATデバイスプロトコル

EtherCAT®

ESM

EtherCATステートマシン(ESM)
EtherCATステートマシンは各EtherCATスレーブデバイスに対するステップバイステップの手順を定義し、実装されている機能に設定を反映します。デバイスはマスターによる状態リクエストを拒否でき、エラーを(AL Statusレジスタのエラーフラグで)通知し、対応するエラーコード(AL Status Codeレジスタ)を付加します。状態リクエスト、状態レスポンスとエラーレスポンスはESC AL Controlレジスタ(reg0x0120)、AL Statusレジスタ(reg0x0130)およびAL Status Codeレジスタ(reg0x0134)を介して行います。ESMはスレーブのビヘイビアを規定し、マスターはこれに応じて対応します。下のESMの説明は各状態遷移の基本動作の概要であり、マスターが送信するコマンドと接続したデバイスの(ローカルの)ビヘイビアを示しています。

用語



INIT (reg0x0120/reg0x0130 = 0x1)**:** ESCレジスタにのみアクセス可能

INIT → PRE-OPERATIONAL

マスター
- クリアー: 設定レジスタ(例):
  - FMMU (reg0x0600 – 0x06FF)
  - SyncManager (SM) (reg0x0800 – 0x087F)
- 設定: 固定物理アドレス(ノードアドレス) (reg0x0010)
  - SM0 for Mailbox 出力 (reg0x0800 – 0x0807)
  - SM1 for Mailbox 入力 (reg0x0808 – 0x080F)
- DC有効時: DC System Time 設定:
  - 伝送遅延補償 (reg0x0900, reg0x0928)
  - オフセット補償 (reg0x0918, reg0x0920)
  - 静的ドリフト補償 (∼15.000 回) (reg0x0910)

スレーブ
- 検証: Mailbox SyncManager 設定
- 開始: Mailbox 通信開始 (START_MBX_HANDLER)
- 応答: PRE-OPERATIONALに状態遷移完了

→ INIT

マスター
- クリアー: SM0 for Mailbox 出力 (reg0x0800 – 0x0807)
  - SM1 for Mailbox 入力 (reg0x0808 – 0x080F)

Slave
- 停止: Mailbox 通信 (STOP_MBX_HANDLER)

PRE-OPERATIONAL (reg0x0120/reg0x0130 = 0x2)**:** Mailbox 通信可能　　**BOOT** (0x3)**: optional**

PRE-OPERATIONAL → SAFE-OPERATIONAL

マスター
- 設定: SDO経由で設定オブジェクトを設定:
  - RxPDO / TxPDO Assignment (obj0x1C12 / obj0x1C13)
  - RxPDO / TxPDO Mapping (reg0x1A00 – 0x1BFF / reg0x1600 – 0x17FF)
  - SM2 出力用 (reg0x0810 – 0x0817)
  - SM3 入力用 (reg0x0818 – 0x081F)
  - FMMU0 (出力用) (reg0x0600 – 0x060F)
  - FMMU1 (入力用) (reg0x0610 – 0x061F)
- DC有効時: SYNC/LATCH ユニット設定:
  - SYNC cycle time 設定 (reg0x09A0 – 0x09A7)
  - DC start time 設定 (reg0x0990 – 0x0997)
  - DC SYNC OUT ユニット設定 (reg0x0980 – 0x0981)
  - DC LATCH IN ユニット設定 (reg0x09A8 – 0x09A9)
  - 連続ドリフト補償開始 (reg0x0910 – 0x0917)
- 開始: 周期プロセスデータ

スレーブ
- 検証: プロセスデータ SyncManager settings, PDO Mapping/Assignment
- 開始: 入力更新 (START_INPUT_HANDLER)
- 動作: 入力有効化
- 応答: SAFE-OPERATIONALに状態遷移完了

→ PRE-OPERATIONAL / INIT

マスター
- クリアー: SM2 出力用 (reg0x0810 – 0x0817)
  - SM3 入力用 (reg0x0818 – 0x081F)
  - FMMU0 (出力用) (reg0x0600 – 0x060F)
  - FMMU1 (入力用) (reg0x0610 – 0x061F)
- 無効化: ディストリビュートクロック (reg0x0980)

スレーブ
- 停止: 周期プロセスデータ

SAFE-OPERATIONAL (reg0x0120/reg0x0130 = 0x4)**:** プロセスデータ通信を開始。まだ出力は安全状態。

SAFE-OPERATIONAL → OPERATIONAL

マスター
- 動作: 出力有効化

スレーブ
- 検証: DCによる時刻同期 (DC有効時)
- 開始: 出力更新 (START_OUTPUT_HANDLER)
- 応答: OPERATIONALに状態遷移完了

→ SAFE-OPERATIONAL / PRE-OPERATIONAL / INIT

スレーブ
- 停止: 出力更新 (STOP_OUTPUT_HANDLER)

OPERATIONAL (reg0x0120/reg0x0130 = 0x8)**:** 入力および出力プロセスデータ有効

状態遷移/デフォルトタイムアウト: PreopTimeout (3000ms)　SafeopOPTimeout (10000ms)　BackToSafeopTimeout (200ms)　BackToInitTimeout (5000ms)

References: ETG.1000.5 – AR (Application Relationship) | ETG.1000.4 – Attributes | ETG.1000.6 – AL state machine | ETG.2000 – InfoType

CoE

CoE
CoEオブジェクトディクショナリ(OD)
ODはCANopen®をベースとし、あらゆるEtherCATデバイスのデータ構造で使用します。各オブジェクトは16bitのインデックスで指定し、8bitのサブインデックス(SI)で指定するオブジェクトエントリーを最大255個まで定義できます。オブジェクトコーディングは以下の3タイプあります。
- VARIABLE: 変数型、SI0のみ
- ARRAY*: 配列型、SI0以外の全てのエントリーが同じデータ型、同じ名称
- RECORD*: 構造体型、各エントリーが異なるデータ型、異なる名称を使用可

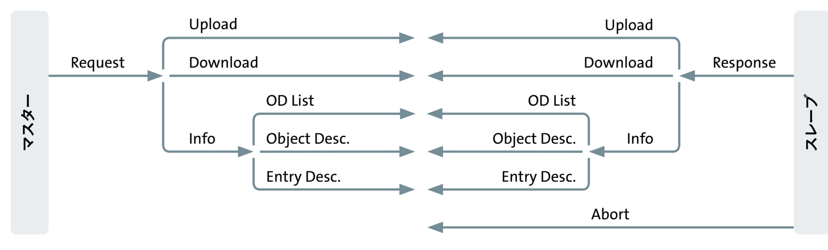
*SI0は実装されたサブインデックスの最大値。SI1は常に16bitアライメントでオフセット。

CoE 基本データタイプ (抜粋)

番号	データ型	番号	データ型	番号	データ型	番号	データ型
0x0001	BOOL/BIT	0x0006	UINT	0x000B	ARRAY of UINT	0x001F	WORD
0x0002	SINT	0x0007	UDINT	0x0011	LREAL	0x0020	DWORD
0x0003	INT	0x0008	REAL	0x0015	LINT	0x0260	ARRAY of INT
0x0004	DINT	0x0009	STRING(n)	0x001B	ULINT	0x0261	ARRAY of SINT
0x0005	USINT	0x000A	ARRAY of BYTE	0x001E	BYTE	0x0262	ARRAY of DINT

Reference: ETG.1020 – Base Data Types

CoE SDO (サービスデータオブジェクト) サービス



SDOサービスはスレーブのオンラインODに対しリードおよびライトアクセスを行うために決められた手順です。これはOD構成をリードするのにも使用されます。マスターはSDOリクエストを、スレーブはSDOレスポンスまたはSDO Abortを送信します。以下のSDOリクエスト/レスポンスサービスが定義されています。
- SDO Upload: オンラインODのオブジェクト/オブジェクトエントリーをリード
- SDO Download: オンラインODのオブジェクト/オブジェクトエントリーをライト
- Get OD List: オンラインODの全オブジェクトインデックスを取得
- Get Object Description: オブジェクト名/コード、データ型、エントリーの最大番号を取得
- Get Entry Description: エントリーの値情報、データ型、ビット長、アクセス権を取得

Reference: ETG.1000.5 – CoE service specification | ETG.1000.6 – CoE coding

SDO コマンド指定子	SDO サービス 0x02: SDO Request	0x03: SDO Response	0x08: SDO Info
0x00	Download Segmented	Upload Segmented	
0x01	Download	Download Segmented	Get OD List Req.
0x02	Upload	Upload	Get OD List Resp.
0x03	Upload Segmented	Download	Get Object Description Req.
0x04	Abort Transfer		Get Object Description Resp.
0x05			Get Entry Description Req.
0x06			Get Entry Description Resp.
0x07			SDO Info Error Req.

モジュラーデバイスプロファイル(MDP)
MDPはあらゆるEtherCATスレーブが物理または論理モジュールをデバイス内に定義するための基本構成であり、CoE ODをベースとしています。スレーブのデータは物理構成and/or論理/ソフトウェア構成に基づきグループ化されます。MDPの構成はいわゆるモジュールに基づきます。モジュールには以下のようにインデックス範囲を割り当てます。
- 1 RxPDO, 1 TxPDO (PdoIncrement=1)
- 各機能のインデックスエリア(入力、出力、設定、情報、診断)ごとに16オブジェクト(IndexIncrement=16)

特定のモジュールに依存しない情報は0xFxxx Device Indexエリアに定義されています。

CoE オブジェクトディクショナリ(OD)構成(MDP構成を含む)
一般的なOD 構成およびMDP 構成(0x6000以降)のインデックス範囲:

MDPデバイス オブジェクトディクショナリ	モジュール 0	モジュール 1	...	モジュール n
通信エリア (0x1000 – 0x1FFF)				
例 obj0x1000, 0x1018, 0x10F3				
RxPDOs (0x1600 – 0x17FF)	0x1600	0x1601	...	0x16nn
TxPDOs (0x1A00 – 0x1BFF)	0x1A00	0x1A01	...	0x1Ann
ベンダー固有エリア (0x2000 – 0x5FFF)				
入力変数エリア (0x6000 – 0x6FFF) TxPODにマッピング可、リードオンリー	0x6000 – 0x600F	0x6010 – 0x601F	...	0x6nn0 – 0x6nnF
出力変数エリア (0x7000 – 0x7FFF) RxPODにマッピング可、リード/ライト可	0x7000 – 0x700F	0x7010 – 0x701F	...	0x7nn0 – 0x7nnF
設定エリア (0x8000 – 0x8FFF) リード/ライト可、通常PODマッピング不可	0x8000 – 0x800F	0x8010 – 0x801F	...	0x8nn0 – 0x8nnF
情報エリア (0x9000 – 0x9FFF) リードオンリー、通常PODマッピング不可	0x9000 – 0x900F	0x9010 – 0x901F	...	0x9nn0 – 0x9nnF
診断エリア (0xA000 – 0xAFFF)	0xA000 – 0xA00F	0xA010 – 0xA01F	...	0xAnn0 – 0xAnnF
デバイスエリア (0xF000 – 0xFFFF)				
例 obj0xF000, 0xF010, 0xF030, 0xF050				

Reference: ETG.5001 – MDP Device model

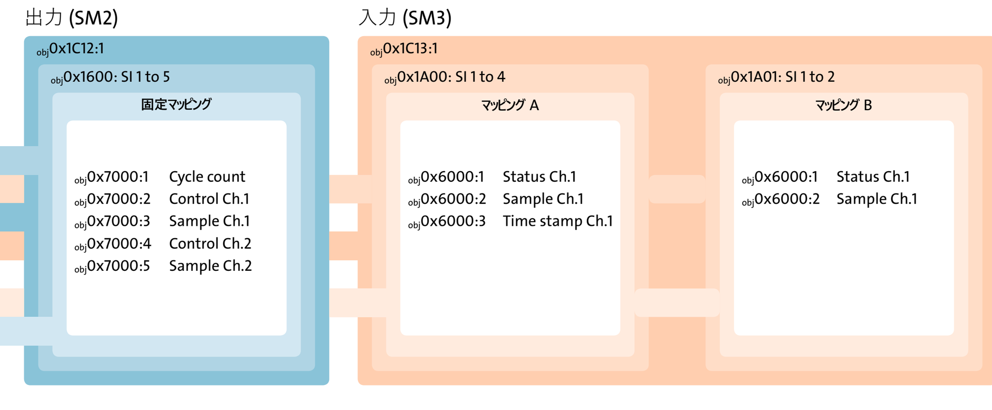
デバイスプロファイル
デバイスプロファイルの大部分はCoE OD(主にインデックス範囲0x6000以降)の構成の定義であり、MDPベースやIEC61800 (CiA402)ドライブプロファイルなどがあります。32ビットのプロファイル番号はスレーブ内のobj0x1000, obj0xF010およびESIエレメント Device:ProfileまたはModule:Profileに記述します。ビット0-15はデバイスプロファイル番号、ビット16-31はサブプロファイル番号で、例えばSEMIデバイスプロファイルのマスフローコントローラーはそれぞれ5003, 2020です。主要なプロファイルを以下に示します。
- ETG.5001.1 – 汎用MDPデバイスモデル仕様書
  あらゆるEtherCATスレーブの基本構成
- ETG.5001.3 – MDPフィールドバスゲートウェイプロファイル仕様書
  EtherCATマスター、Profibus DP、CAN、CANopen、DeviceNetなど
- ETG.5001.4 – MDPセーフティモジュール仕様書
  FSoEデジタルI/Oコネクション、FSoEセーフティドライブプロファイル、FSoEマスター
- ETG.5003 – SEMIデバイスプロファイル
  MDP構成に準拠。マスフローコントローラー、温調、圧力ゲージ、バルブ、チラー、ポンプ、RF DC電源など
- ETG.6010 – CiA402ドライブプロファイル(IEC61800-7-201)の実装規定

プロセスデータ

プロセスデータ設定
EtherCATプロセスデータ設定は非常に柔軟なPDO構成を実現します。PDO設定は固定、選択可能、編集可能のいずれかです。入力変数(obj0x6000 – 0x6FFF)および出力変数(obj0x7000 – 0x7FFF)はPDO Mappingオブジェクト(RxPDO Mapping: obj0x1600 – 0x17ff, TxPDO Mapping: obj0x1A00 – 0x1BFF)に構成を定義し、SMに対応したPDO Assignment obj0x1C12(SM2), obj0x1C13(SM3)に割り付けます。PDO MappingおよびPDO Assignmentオブジェクトは、コンプレックスデバイスではオンライン/オフラインODの両方で使用し、シンプルデバイスではEtherCATスレーブ情報ファイルで概念的に使用します。

例: PDO Mapping と PDO Assignment
この例は固定出力マッピングを示しています。入力マッピングは選択可能です。マッピングAまたはBをobj0x1C13で SM3に割り当てできます。

OD エリア	OD インデックス範囲
通信エリア	0x1000 – 0x1FFF
RxPDO Mapping	0x1600 – 0x17FF
TxPDO Mapping	0x1A00 – 0x1BFF
RxPDO Assignment	0x1C12
TxPDO Assignment	0x1C13
モジュールエリア	0x6000 – 0xAFFF
入力変数エリア	0x6000 – 0x6FFF
出力変数エリア	0x7000 – 0x7FFF
デバイスエリア	0xF000 – 0xFFFF



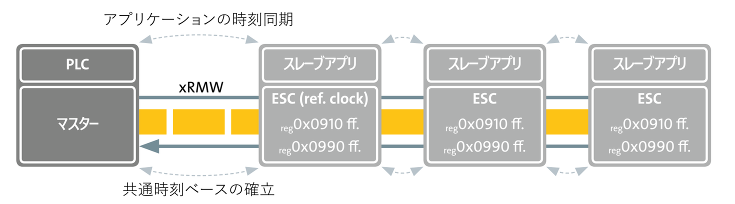
References: ETG.1000.5 – Process data interaction | ETG.1000.6 – Object Dictionary | ETG.5001.1 – PDO Mapping and PDO Assign

DC ユニットと時刻同期

ディストリビュートクロックと時刻同期
マスターとスレーブアプリケーション間の同期はネットワーク内の共通の時刻(DC System Time)をベースとします。同期モードはこの共通の時刻を使ってローカルアプリケーションを同期する方法を定義します。SYNC/LATCHユニットはDC System Timeを基準としてSYNC/LATCHイベントを生成します。

DC System Time
DC System Timeは2000年1月1日0時0分を起点とするナノ秒単位の64ビットの時刻もしくは32ビットの時刻です。System Timeの設定後、マスターとスレーブは同じ基準の時刻を共有します。マスターからみて最初のDCスレーブの時刻がリファレンスクロックとして使用されます。各スレーブデバイスはSystem Timeのローカルコピーをreg0x0910に保持します。

DC System Time 設定 (1-3) と動的ドリフト補償 (4)



1. **伝送遅延**: reg0x0900へのライトで全ポートのフレーム受信時刻をラッチ。ラッチした時刻をリード。遅延を計算。各DCスレーブのreg0x0928に遅延時間をライト。
2. **オフセット**: 個々のローカルタイムをリード。リファレンスクロックとのオフセットを計算。個々のreg0x0920にオフセット値をライト。
3. **静的ドリフト補償**: 複数回(∼15,000) xRMWデータグラムでリファレンスクロックのSystem Timeをリードして各DCスレーブにライト。
4. **動的ドリフト補償**: 周期フレームでxRMWによりSystem Timeを配信。数msのサイクルタイムごとにSystem Timeの偏差を維持。

Reference: ETG.1000.4 – Distributed clock

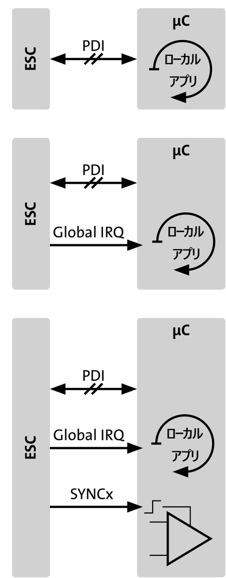
Reference: ETG.1020 – Synchronization

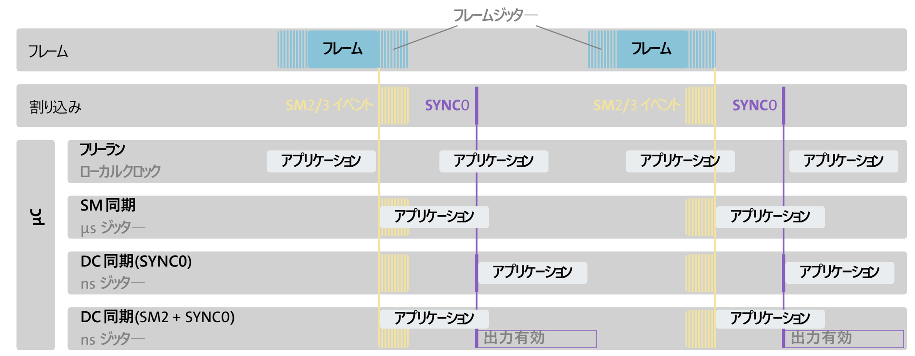
同期モード
アプリケーションによって必要とされる同期のレベルは異なり、EtherCATには様々な同期モードが用意されています。DCモードの基本動作はreg0x0980- 0x0981で設定します。その他の関連情報はSyncManager Parameterオブジェクトobj0x1C32(SM2)およびobj0x1C33(SM3)で参照でき、最小サイクルタイム、演算&コピー時間、エラーカウンターなどもあります。

フリーラン: アプリケーションはローカルクロックでトリガーされ、EtherCAT通信サイクルとは独立して実行されます。ESIエレメントDevice:Dcは記述せず、フリーランだけをサポートする場合obj0x1C32/3はオプションです。

SM同期: アプリケーションはSM2(SM3)イベントに同期します。このイベントはプロセスデータがSM2にライト(またはSM3からリード)されたときに生成されます。イベントはグローバルIRQにマッピングされるか、reg0x0220をポーリングして取得します。SM同期のみをサポートする場合ESIエレメントDevice:Dcは記述しません。SM同期とDC同期の両方をサポートする場合、SM同期はESIでDC:AssignActivate="#x0000"と記述します。

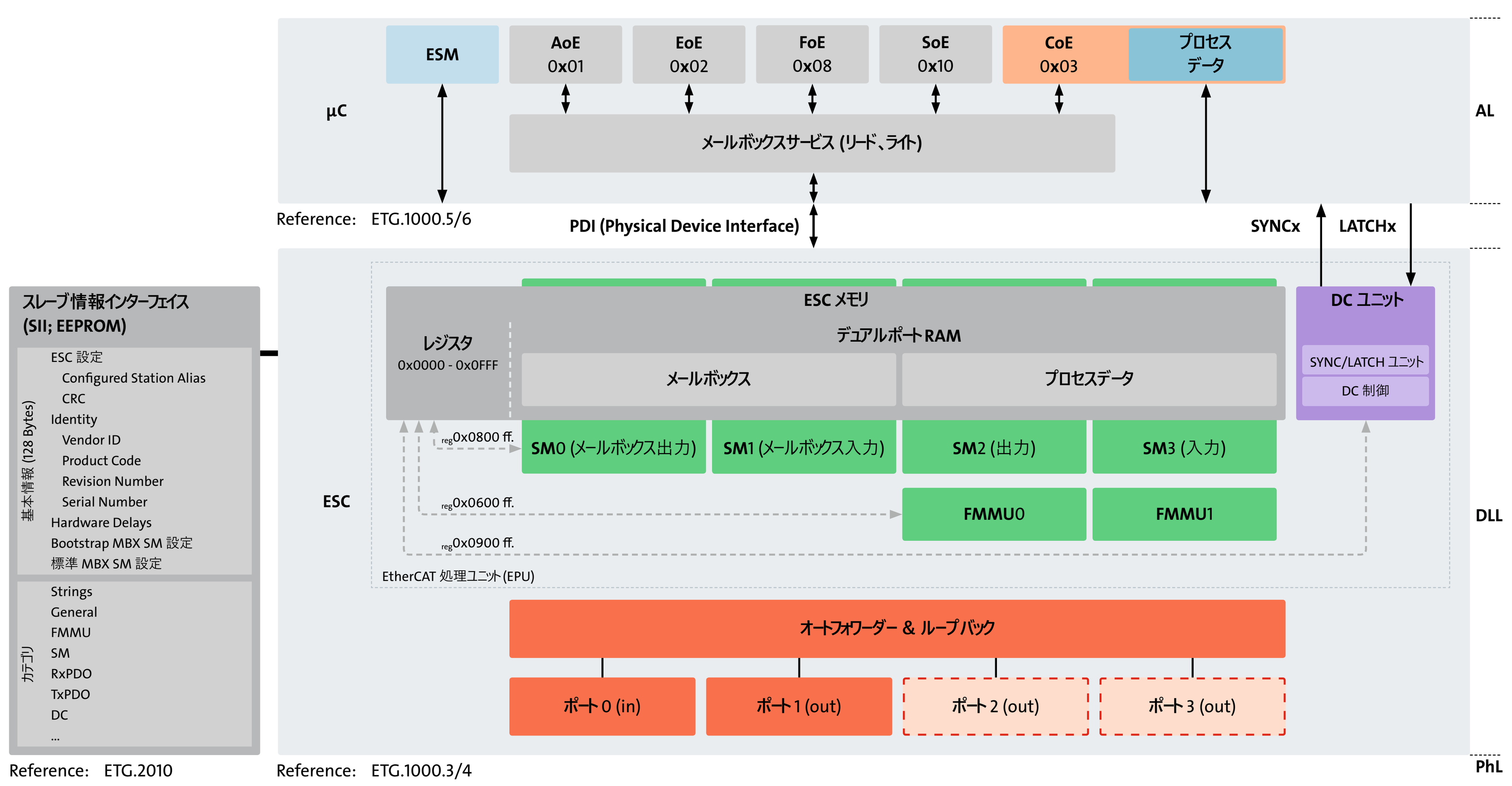
DC同期: アプリケーションはDCベースの割込み信号(SYNC0, SYNC1; ナノ秒レベル)で同期し、DCユニットがSYNC0/1割込みを生成します。各種あるDCモードで次の2点が基本概念です。
- SYNC0: この割込みでローカルサイクルの処理をトリガー
- SM2 & SYNC0 (出力イベントの同期精度を向上): SM2イベントでSM2から出力データのリードをトリガーし演算処理後にハードウェア出力ドライバーに値をライト、SYNC0イベントで出力ドライバーの有効化をトリガー。ESIエレメントでDC:AssignActivate="#x0300"と記述し、SYNC0イベントを生成。





Reference: ETG.1020 – Synchronization | ETG.2000 – DC

スレーブ内部構造とEtherCATスレーブコントローラ(ESC)
ESCはハードウェアによりEtherCATフレームをオンザフライで処理し、EtherCATのデータリンク層(DLL)の機能が実装されています。これにはSyncManager(SM)、フィールドバスメモリ管理ユニット(FMMU)やDCユニットなどがあります。一般的にEtherCATスレーブはESM、パラメータ操作やプロセスデータ処理のようなアプリケーション層(AL)の機能ははμC内のソフトウェアとして実装します。この形態のスレーブを「コンプレックスデバイス」といいます。パラメータ設定がなくALエラー処理が不要であるような非常に単純なI/OデバイスはESCのデジタルI/Oインタフェースに直接I/Oハードウェアドライバを接続します。このようなスレーブを「シンプルデバイス」といいます。
ブロック図はスレーブのハードウェア構成、ESC内の機能名称、ソフトウェア構成とプロトコルを示しています。



SM & FMMU

SyncManager (SM)
SMはEtherCAT(マスター)とPDI(μC)の両側からのESCメモリーアクセスを管理し、データのコンシステンシーを保証します。メールボックス通信ではメールボックスのメッセージが上書きされないように保護します(1-バッファモード)。プロセスデータ通信では常にEtherCAT側からのライトが可能なメモリとPDI側からのリードが可能な最新のライト済みメモリを保証します(3-バッファモード)。その逆方向も同様です。SyncManager 2/3の長さはRx/TxPDOの長さと同じであり、SMはライト/リードアクセスに対して3つのバッファから常に適切なメモリ領域を割り当てます。

Reference: ETG.1000.4 – Sync manager

フィールドバスメモリ管理ユニット(FMMU: Fieldbus Memory Management Units)
プロセスデータ交換には論理アドレスコマンド(LRD, LWR, LRW)を使用します。1つのLxxコマンドは1台以上の複数スレーブをアドレス指定します。各スレーブのFMMUは起動時に設定され、EtherCATコマンドの論理アドレス空間とESCの物理メモリ間の割り当てをします。FMMUの設定はreg0x0600から始まるレジスタで行います。詳しくは例3を参照してください。

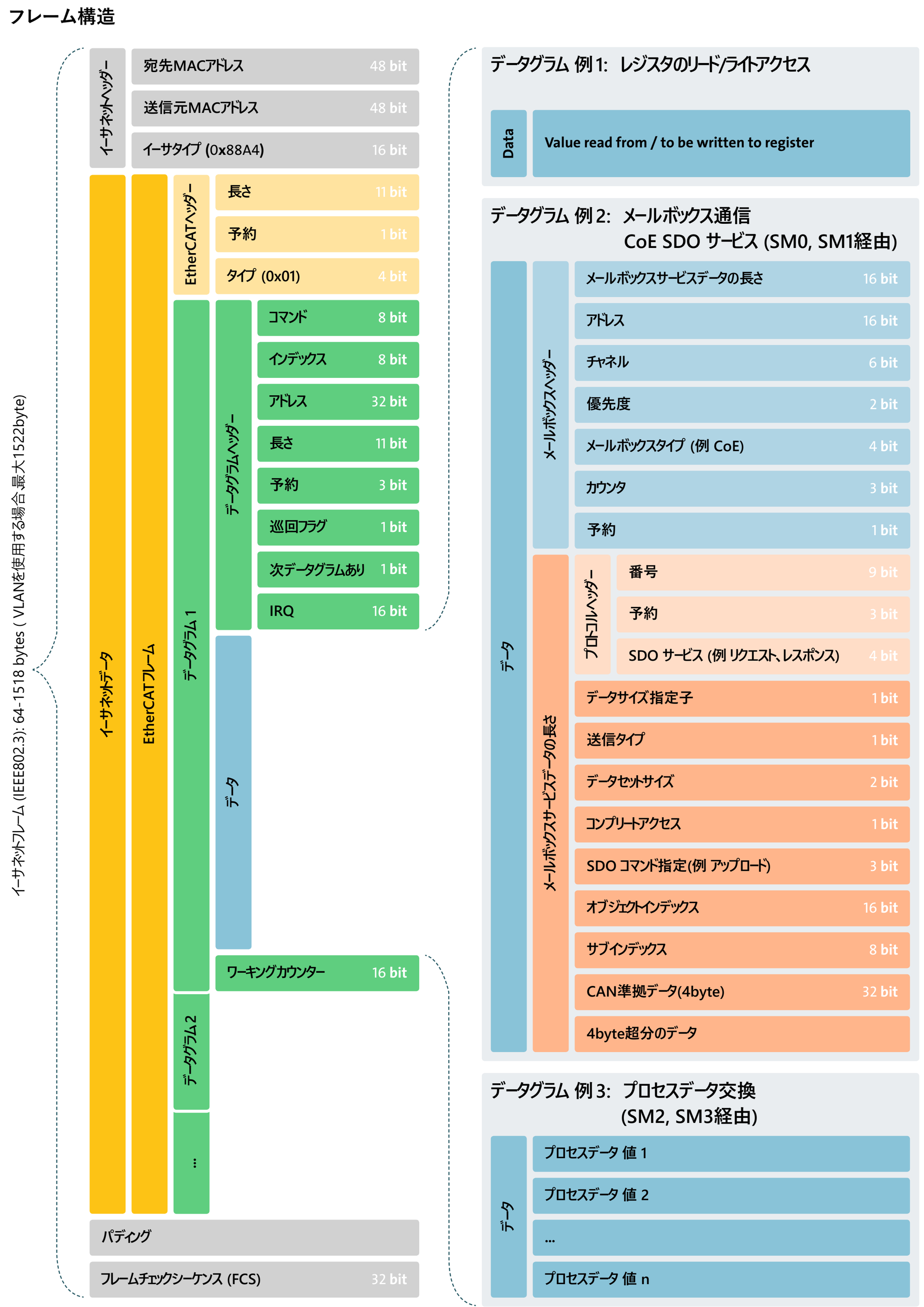
Reference: ETG.1000.4 – Fieldbus memory management unit

フレーム処理と例

通信のしくみ
EtherCAT通信は常にマスターがそのイーサネットインタフェースからフレームを送信して開始します。フレームはESCによりオンザフライで処理されます。ESC内のルーティングは「ランドアバウト」方式で行われます。EtherCAT処理ユニット(EPU)の通過後、フレームは次のポートにフォワードされ、ポートがオープン状態であればフレームは次のスレーブに送出されます。 戻りのフレームはポート0からマスターの方向に返送されます。ポート0は常にスレーブのINポートとして使用します。トポロジーは必ず論理的にリングを形成し、フレームの衝突や輻輳は原理上起こりえません。通信スループットは正確に計算でき、エラーはステータスレジスタやエラーカウンターレジスタ(reg0x0100, reg0x0300 – 0x0313)で簡単に検出できます。

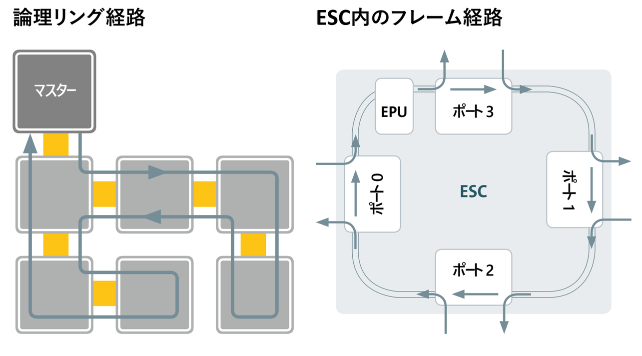
Reference: ETG.1000.4 – Frame processing principles

フレーム構造



References: ETG.1000.4 – EtherCAT frame structure | ETG.1000.6 – CoE coding

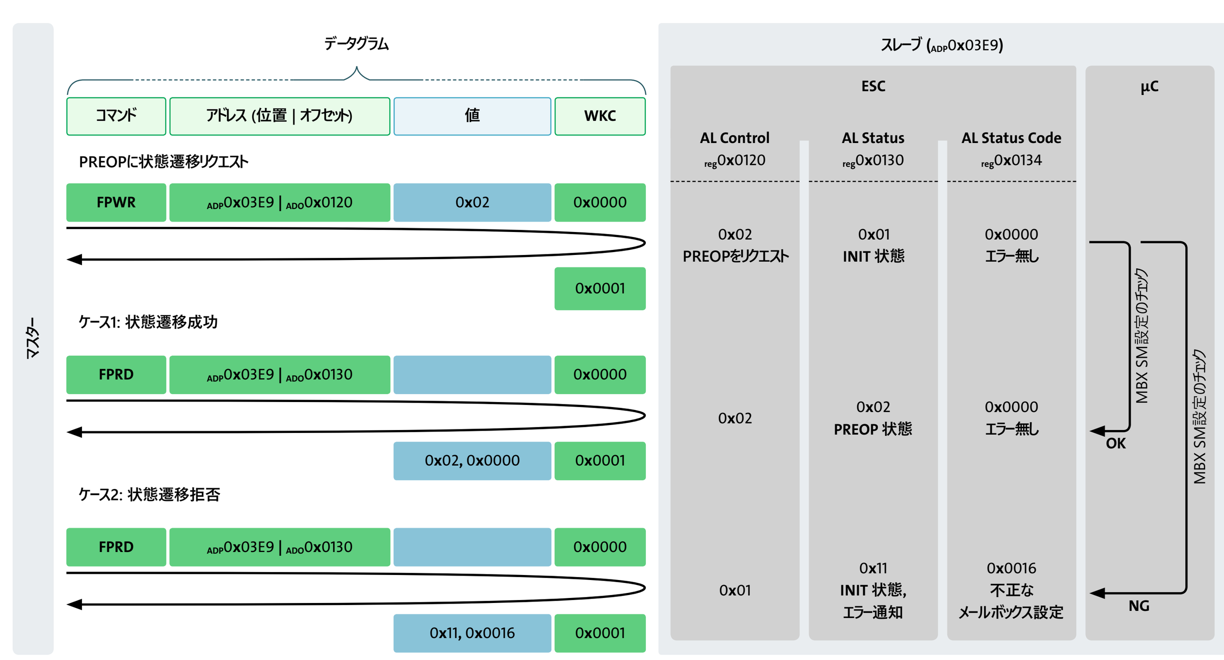
論理リング経路　**ESC内のフレーム経路**



ポートとエラーレジスタ

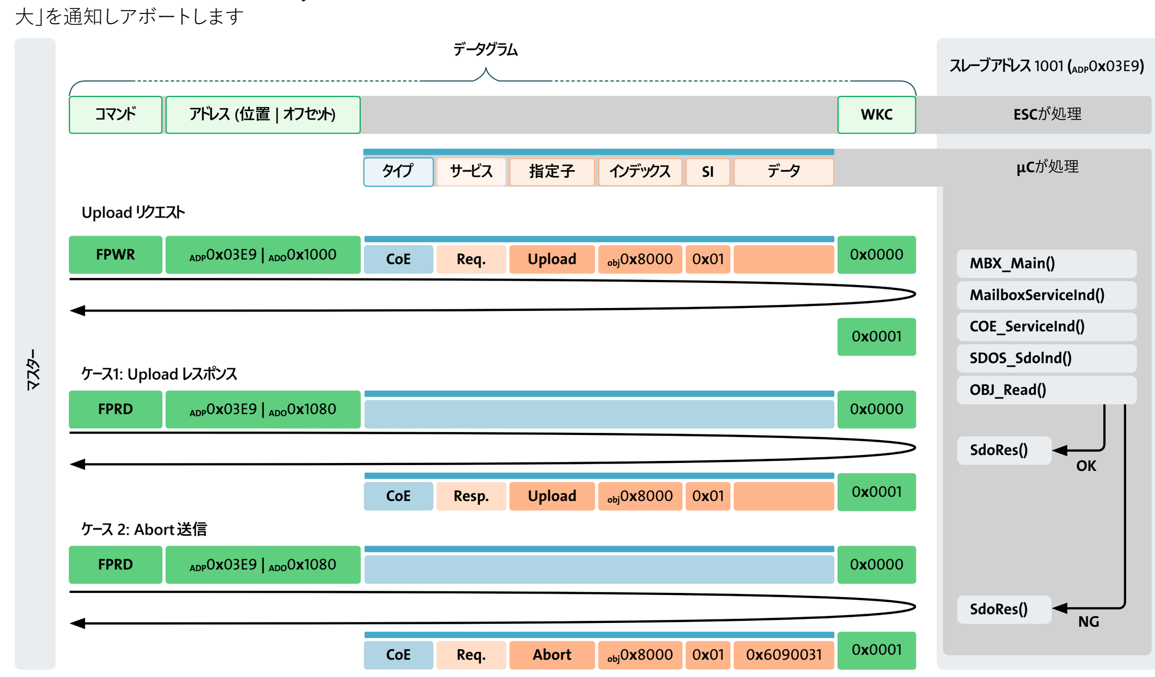
レジスタ	ポート 0	1	2	3
DL Control	reg0x0100 – 0x0103			
DL Status	reg0x0110 – 0x0111			
フレームエラーカウンター	reg0x0300	reg0x0302	reg0x0304	reg0x0306
RXエラーカウンター	reg0x0301	reg0x0303	reg0x0305	reg0x0307
リンクロストカウンター	reg0x0310	reg0x0311	reg0x0312	reg0x0313

データグラム 例1: レジスタのリード/ライトアクセス
状態遷移時のESM関連レジスタ0x0120, 0x0130, 0x0134のリード/ライトアクセスの例



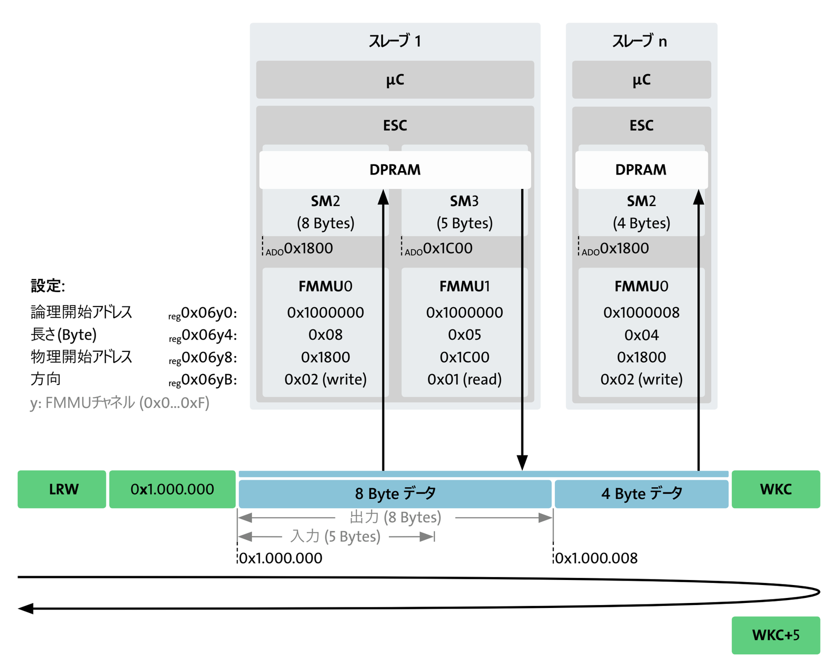
Reference: ETG.1000.6 – Protocol State Machine

データグラム 例1: メールボックス通信、CoE SDOサービス (SM0, SM1経由)
この例ではSDO DownloadリクエストをSM0メールボックス出力にライトし、そのレスポンス/アボートをSM1メールボックス入力からリードします。ケース1は設定オブジェクトobj0x8000:01に値のダウンロードが成功し、ケース2はAbort Code 0x06090031「書き込むパラメータ値が過大」を通知しアボートします



Reference: ETG.1000.4 – EtherCAT frame structure | ETG.1000.6 – SDO

データグラム 例3: プロセスデータ交換 (FMMU)
この例でプロセスデータを論理リード・ライトコマンド(LRW)を使用した周期交換です。スレーブのDPRAM上のSM2/3バッファと論理アドレスの対応付けはFMMUで行います。



Reference: ETG.1000.4 – Fieldbus memory management unit

コマンド
EtherCATコマンド(データグラム)は1台以上のスレーブをアドレス指定します。ノードアドレス指定(APxx, FPxx)、論理アドレス指定(Lxx)とブロードキャスト(Bxx)が用意されています。リード/ライト操作に成功したスレーブはワーキングカウンター(WKC)をインクリメントします。

指定子	コマンド	説明
0x00	NOP	何もしない
0x01	APRD	オートインクリメント物理リード
0x02	APWR	オートインクリメント物理ライト
0x03	APRW	オートインクリメント物理リード&ライト
0x04	FPRD	設定アドレス物理リード
0x05	FPWR	設定アドレス物理ライト
0x06	FPRW	設定アドレス物理リード&ライト
0x07	BRD	ブロードキャストリード
0x08	BWR	ブロードキャストライト
0x09	BRW	ブロードキャストリード&ライト
0x0A	LRD	論理アドレスリード
0x0B	LWR	論理アドレスライト
0x0C	LRW	論理アドレスリード&ライト
0x0D	ARMW	オートインクリメント物理リード&マルチライト
0x0E	FRMW	設定アドレス物理リード&マルチライト

ワーキングカウンター

コマンド	動作結果	インクリメント値
xRD	リード成功	+1
xWR	ライト成功	+1
xRW	リード成功 ライト成功	+1 +2

Reference: ETG.1000.4 – EtherCAT frame structure | ETG.1000.4 – Error detection overview