

取扱説明書

EL6695

EtherCAT ブリッジターミナル

バージョン: 1.4.1
日付: 2020-02-04

BECKHOFF

目次

1	序文	5
1.1	取扱説明書に関する注記	5
1.2	安全に関する指示事項	5
1.3	取扱説明書の改訂履歴	6
1.4	EtherCATデバイスのバージョン識別	7
1.4.1	ベッコフ識別コード(BIC)	11
2	製品概要	13
2.1	概要	13
2.2	技術データ	13
2.2.1	ベッコフEtherCATデータ交換デバイスの比較	14
2.3	設置に関する動作条件	15
3	基本的な通信	16
3.1	EtherCATの基本	16
3.2	EtherCATの配線 - 結線	16
3.3	ウォッチドッグ設定に関する一般的な注記	17
3.4	EtherCATステートマシン	19
3.5	CoEインターフェイス	20
3.6	ディストリビュートクロック	25
4	取付けおよび配線	26
4.1	ESD保護に関する指示事項	26
4.2	取付けおよび取外し - フロントロック解除式ターミナル	27
4.3	推奨する取付けレール	29
4.4	設置方向	30
4.5	パッシブターミナルの配置	32
4.6	ULに関する注記	33
4.7	接続	34
5	コミッショニング	35
5.1	TwinCATクイックスタート	35
5.1.1	TwinCAT 2	37
5.1.2	TwinCAT 3	46
5.2	TwinCAT開発環境	57
5.2.1	TwinCATリアルタイムドライバのインストール	58
5.2.2	ESIデバイス記述ファイルに関する注記	63
5.2.3	オフラインでのコンフィグレーションの作成	66
5.2.4	オンラインでのコンフィグレーションの作成	71
5.2.5	EtherCATサブスクライバコンフィグレーション	78
5.3	一般的な注意 - EtherCATスレーブアプリケーション	87
5.4	CoEオブジェクトの概要	95
5.4.1	標準CoEオブジェクト	95
5.4.2	ターミナル固有のCoEオブジェクト	96
5.4.3	プロファイル固有のCoEオブジェクト	102
6	機能および動作モード	105
6.1	基本機能の原則	105
6.1.1	TwinCATでのEL6695へのアクセス	106
6.2	EL6692との互換性	106
6.3	ステートマシンEL6695	107
6.4	周期プロセスデータPDO	107
6.4.1	フロー制御	107

6.4.2	対称PDO Mapping	111
6.4.3	選択的PDO Mapping	115
6.4.4	FSoE転送	125
6.5	ローカルメールボックスプロトコル	125
6.5.1	CoE - Can over EtherCAT	125
6.6	送信機能付きのメールボックスプロトコル	126
6.6.1	CoEアクセス中のAoEの使用例	126
6.6.2	EoE - Ethernet over EtherCAT	128
6.6.3	AoE - ADS over EtherCAT	131
6.6.4	FoE - Filetransfer over EtherCAT	133
6.6.5	VoE - vendor-specific protocol over EtherCAT	144
6.6.6	SoE - Servo Drive Profile over EtherCAT	144
6.7	ディストリビュートクロック	144
6.8	オンラインスキャン	147
6.9	EL6692互換モード	147
6.9.1	PDO MappingとAssignによるモジュール動作	147
6.10	EL6695パフォーマンスモード	147
6.10.1	PDO Mappingの基本原則	147
6.10.2	Initコマンドなし	147
6.10.3	オブジェクトディスクリプションのダウンロード	148
6.10.4	デバイスエミュレーション	148
6.11	アプリケーション固有の変数定義	148
7	付録	149
7.1	EtherCAT ALステータスコード	149
7.2	ファームウェアの互換性	149
7.3	ファームウェア更新EL/ES/EM/ELM/EPxxxx	150
7.3.1	デバイスESIファイル/XML	150
7.3.2	ファームウェアの説明	153
7.3.3	コントローラファームウェア*.efwの更新	154
7.3.4	FPGAファームウェア*.rbf	156
7.3.5	複数のEtherCATデバイスの同時更新	160
7.4	ファームウェア更新に関する補足	161
7.5	診断	162
7.6	工場出荷状態の復元	162
7.6.1	一般的なデバイスのリセット	162
7.6.2	工場出荷状態の復元	162
7.7	サポートとサービス	164

1 序文

1.1 取扱説明書に関する注記

対象となる読者

この説明書は関連する国内規格を熟知した、制御およびオートメーションエンジニアリングの専門家の使用のみを目的としています。

本製品のインストールおよびコミッショニングの際は、必ず以下の注意事項と説明に従ってください。
(インストールおよびコミッショニング時点での最新の取扱説明書を参照するようにしてください。)

本製品を使用する上での責任者は、本製品の用途および使用方法が、関連するすべての法律、法規、ガイドラインおよび規格を含む、安全に関するすべての要件を満たしていることを確認してください。

免責事項

この取扱説明書の記載内容は、一般的な製品説明および性能を記載したものであり、場合により記載どおりに動作しないことがあります。

製品の情報・仕様は予告なく変更されます。

この説明書に記載されているデータ、図および説明に基づいて、既に納品されている製品の変更を要求することはできません。掲載されている写真やイラストと、実際の製品は異なる場合があります。この説明書は最新でない可能性があります。必ず<https://infosys.beckhoff.com>に掲載された最新バージョンの説明書を参照してください。

商標

Beckhoff®、TwinCAT®、EtherCAT®、EtherCAT G®、EtherCAT G10®、EtherCAT P®、Safety over EtherCAT®、TwinSAFE®、XFC®、XTS®およびXPlanar®は、Beckhoff Automation GmbHの登録商標です。この取扱説明書で使用されているその他の名称は商標である可能性があり、第三者が独自の目的のために使用すると所有者の権利を侵害する可能性があります。

特許出願

EtherCAT Technologyについては、欧州特許EP1590927、EP1789857、EP1456722およびEP2137893、ドイツ特許DE102015105702に記載されていますが、これらに限定されるものではありません。



EtherCAT®は、Beckhoff Automation GmbHの登録商標および特許技術です。

著作権

© Beckhoff Automation GmbH & Co. KG, Germany.

明示的な許可なく、本書の複製、配布、使用、および他への内容の転載は禁止されています。これに違反した者は損害賠償の責任を負います。すべての権利は、特許、実用新案、意匠の付与の際に留保されます。

1.2 安全に関する指示事項

安全に関する注意事項

この取扱説明書に記載された安全に関する指示や注意事項はよくお読みになり、必ず指示に従ってください。

製品ごとの安全に関する指示事項は、以下のページ、または取り付け、配線、コミッショニングなどに関する箇所に記載されています。

免責事項

すべての製品は、用途に適した特定のハードウェア構成およびソフトウェア構成を有する状態で供給されます。ハードウェアまたはソフトウェアに取扱説明書に記載されている以外の変更を加えることは許可されていません。許可されていない変更を加えると、Beckhoff Automation GmbH & Co. KGの保証の対象外となります。

使用者の資格

この説明書は対応する国内法規を熟知した、制御およびオートメーションエンジニアリングの専門家の使用を目的としています。

安全記号の説明

この取扱説明書では、安全に関する指示や注意事項とともに以下の安全記号を使用します。安全に関する指示事項はよくお読みになり、必ず指示に従ってください。

⚠ 危険

重大な人的傷害の危険

この記号が付いた安全に関する注意事項に従わないと、人命および健康に直ちに危害を及ぼします。

⚠ 警告

人的傷害の危険

この記号が付いた安全に関する注意事項に従わないと、人命および健康に危険を及ぼします。

⚠ 注意

人的傷害の恐れ

この記号が付いた安全に関する注意事項に従わないと、人命および健康に危険を及ぼす恐れがあります。

注記

環境汚染/物的損害またはデータ消失の恐れ

この記号が付いた安全に関する注意事項に従わないと、環境汚染、物的損害、またはデータ消失につながる恐れがあります。



ヒントまたはアドバイス

この記号が示す情報により、さらに理解が深まります。

1.3 取扱説明書の改訂履歴

バージョン	コメント
1.4.1	- 構成の更新
1.4.0	- FoEデータスループットの例の補足 - 複数の追加/修正 - 構成の更新 - リビジョンステータスの更新
1.3.0	- セクション「ファンクションおよび動作モード」の更新 (FoE動作モードの補足)
1.2.0	- セクション「概要」の更新 - セクション「ファンクションおよび動作モード」の更新 - チャプタ「接続」の補足 - セクション「LED診断」の更新
1.1.0	- セクション「ファンクションおよび動作モード」の更新 - セクション「技術データ」の更新 - 構成の更新 - リビジョンステータスの更新
1.0.0	- 初版
0.9.8	- セクション「選択的PDO Mapping」に「グローバルデータ型への変換」を追加 (暫定版 - 変更の可能性あり)
0.9.7	- セクション「CoEアクセス用のAoEアプリケーション」を修正
0.9.6	- セクション「ファンクションおよび動作モード」の拡張

バージョン	コメント
0.9.5	- セクション「転送可能なメールボックスプロトコル」にセクション「CoEアクセス用のAoEアプリケーション」を挿入
0.9	- 初回製品シリーズ向けの暫定版取扱説明書
0.1	- 初版作成

1.4 EtherCATデバイスのバージョン識別

名称

ベッコフEtherCATデバイスには、以下で構成する14桁の名称があります。

- ・ ファミリーキー
- ・ タイプ
- ・ バージョン
- ・ リビジョン

例	ファミリー	タイプ	バージョン	リビジョン
EL3314-0000-0016	ELターミナル (12 mm、ケーブル接続不要)	3314 (4チャンネル熱電対ターミナル)	0000 (基本タイプ)	0016
ES3602-0010-0017	ESターミナル (12 mm、プラグ着脱可能な接続レベル)	3602 (2チャンネル電圧計測)	0010 (高精度バージョン)	0017
CU2008-0000-0000	CUデバイス	2008 (8ポートファーストイーサネットスイッチ)	0000 (基本タイプ)	0000

注記

- ・ 前述の要素が、**技術的な名称**となります。以下では、EL3314-0000-0016を例としています。
- ・ EL3314-0000はオーダー識別子であり、通常「-0000」の場合はEL3314に省略されます。「-0016」はEtherCATリビジョンです。
- ・ **オーダー識別子**は以下で構成されます。
 - ファミリーキー (EL、EP、CU、ES、KL、CXなど)
 - タイプ (3314)
 - バージョン (-0000)
- ・ **リビジョン「-0016」**は、EtherCAT通信に関する機能拡張のような技術的な更新を示しており、ベッコフが管理しています。原則として、取扱説明書などに記載のない限り、上位リビジョンのデバイスで下位リビジョンのデバイスを置換できます。各リビジョンの関連事項や同一機能については、通常XML形式の記述ファイル (ESI、EtherCAT Slave Information) が用意されており、ベッコフのWebサイトからダウンロードできます。2014年1月から、リビジョンがIP20ターミナルの外側に記載されるようになりました。図、「**バッチ番号およびリビジョンID (2014年1月以降)**」が記載された**EL5021 ELターミナル、標準IP20 I/Oデバイス**を参照してください。
- ・ タイプ、バージョン、およびリビジョンは内部的には16進数で保存されていますが、10進数で表記されます。

識別番号

ベッコフEtherCATデバイスには、ラインごとに異なる識別番号が付けられています。

製造ロット/バッチ番号/シリアル番号/日付コード/D番号

通常、ベッコフI/Oデバイスのシリアル番号は、デバイスまたはステッカーに印字された8桁の数字です。シリアル番号は納品時の状態のコンフィギュレーションを表しているため、バッチの個々のモジュールを区別せずに、製造バッチ全体を示しています。

シリアル番号の構成: KK YY FF HH

KK - 製造された週 (CW、暦週)
 YY - 製造された年
 FF - ファームウェアバージョン
 HH - ハードウェアバージョン

例

シリアル番号: 12063A02: 12 - 製造された週 CW12、06 - 製造された年 2006年、3A - ファームウェアバージョン3A、02 - ハードウェアバージョン02

IP67対応デバイスは例外的に、以下の構文が使用されます(各デバイスの取扱説明書を参照)。

構文: D ww yy x y z u

D - 名称のプレフィックス

ww - 暦週

yy - 年

x - バスPCBのファームウェアバージョン

y - バスPCBのハードウェアバージョン

z - I/O PCBのファームウェアバージョン

u - I/O PCBのハードウェアバージョン

例: D.22081501 : 2008年のCW22、バスPCBのファームウェアバージョン: 1、バスPCBのハードウェアバージョン: 5、I/O PCBのファームウェアバージョン: 0 (このPCBにはファームウェア不要)、I/O PCBのハードウェアバージョン: 1

固有のシリアル番号/ID、ID番号

さらに、シリーズによっては個々のモジュールに一意となる固有のシリアル番号が付けられています。

該当するその他の取扱説明書も参照してください。

- ・ IP67: [EtherCATボックス](#)
- ・ セーフティ: [TwinSAFE](#)
- ・ 製造工場の校正証明書付きターミナルおよびその他の計測用ターミナル

マーキングの例



図 1: シリアル/バッチ番号、およびリビジョンIDが記載されたEL5021 ELターミナル、標準IP20 I0デバイス(2014年1月以降の印字)



図 2: シリアル/バッチ番号が記載されたEK1100 EtherCATカプラ、標準IP20 I0デバイス



図 3: シリアル/バッチ番号が記載されたCU2016スイッチ

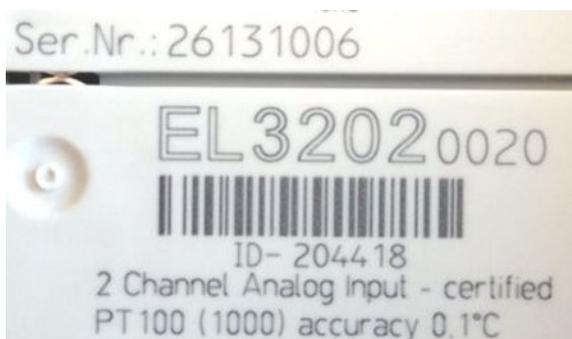


図 4: シリアル/バッチ番号26131006および固有のID番号204418が記載されたEL3202-0020

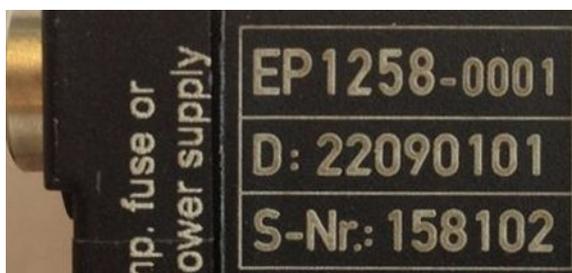


図 5: バッチ番号/日付コード22090101および固有のシリアル番号158102が記載されたEP1258-00001 IP67 EtherCATボックス



図 6: バッチ番号/日付コード071201FFおよび固有のシリアル番号00346070が記載されたEP1908-0002 IP67 EtherCAT安全ボックス



図 7: バッチ番号/日付コード50110302および固有のシリアル番号00331701が記載されたEL2904 IP20安全ターミナル



図 8: 固有のID番号 (QRコード) 100001051およびシリアル/バッチ番号44160201が記載されたELM3604-0002ターミナル

1.4.1 ベックホフ識別コード(BIC)

製品を一意に識別するためのベックホフ識別コード (BIC) が、多くのベックホフ製品に適用され始めています。BICはData Matrixコード(DMC、コードスキームECC200)として表され、その内容はANSI規格MH10.8.2-2016に基づいています。

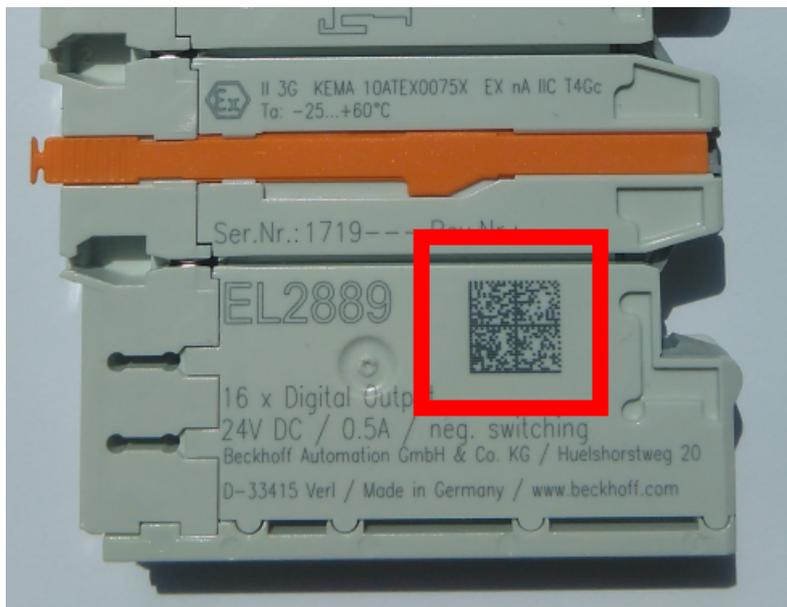


図 9: Data Matrixコードで表す BIC(DMC、コードスキームECC200)

BICはすべての製品グループに順次導入される予定です。

BICは以下のいずれかの場所に記載されています(製品によって異なります)。

- ・ 梱包箱
- ・ 製品(十分なスペースがある場合)
- ・ 梱包箱および製品

機械可読データであるBICは、お客様が製品の取り扱いや管理にも使用できる情報を含んでいます。

それぞれの情報は、いわゆるデータ識別子(ANSI MH10.8.2-2016)を使用して一意に識別できます。データ識別子の後には、文字列が続きます。データ識別子と文字列の最大合計長は、下表のとおりです。情報が短い場合は、スペースが付加されます。1~4のデータは必ず含まれています。

以下の情報が含まれています。

項目番号	情報のタイプ	説明	データ識別子	データ識別子を含む桁数	例
1	ベッコフの注文番号	ベッコフの注文番号	1P	8	<emphas is type="RoteSchr i ft">1P</ emphas is>072222
2	ベッコフトレーサビリティ番号 (BTN)	固有のシリアル番号、下の注記を参照	S	12	<emphas is type="RoteSchr i ft">S</ emphas is>BTnk4p 562d7
3	製品型番	ベッコフ製品型番。EL1008など	1K	32	<emphas is type="RoteSchr i ft">1K</ emphas is>EL1809
4	数量	梱包箱内の数量。1、10など	Q	6	<emphas is type="RoteSchr i ft">Q</ emphas is>1
5	バッチ番号	オプション：製造年および週	2P	14	<emphas is type="RoteSchr i ft">2P</ emphas is>401503 180016
6	ID/シリアル番号	オプション：現行のシリアル番号体系。セーフティ製品など	51S	12	<emphas is type="RoteSchr i ft">51S</ emphas is>678294 104
7	派生タイプ	オプション：標準製品に基づく派生タイプ番号	30P	32	<emphas is type="RoteSchr i ft">30P</ emphas is>F971, 2*K183
...					

その他のタイプの情報およびデータ識別子は、ベッコフが内部処理に使用します。

BICの構造

項目1~4および6の復号情報の例。データ識別子は分かりやすいように赤で表記しています。

BTN

BICの重要な部分は、ベッコフトレーサビリティ番号 (BTN、項目番号2) です。BTNは8文字で構成する固有のシリアル番号です。ベッコフは長期的に他のすべてのシリアル番号体系をBTNに置換していきます (I0コンポーネントのバッチ名称、セーフティ製品の従来のシリアル番号範囲など)。BTNは徐々に導入されるため、BICにBTNがコーディングされていない場合もあります。

注記

この情報は入念に準備されています。ただし、記載されている方式について、継続的にさらなる開発が行われていきます。方式や製品の情報は予告なく変更されます。本取扱説明書内の情報、図、および説明の変更によって不都合が発生しても、当社は責任を負いかねます。

2 製品概要

2.1 概要

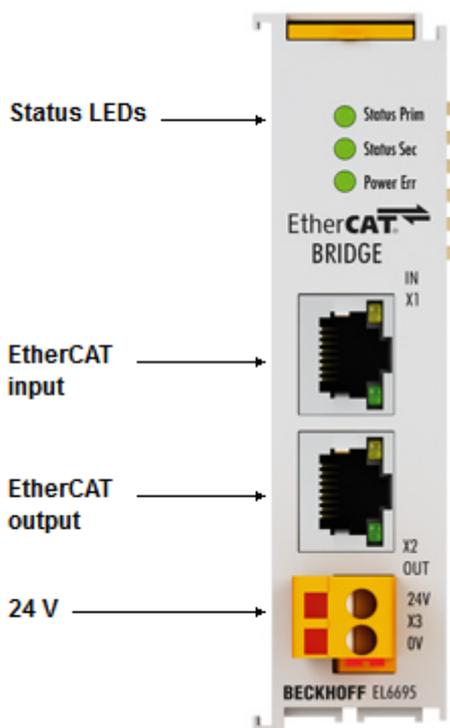


図 10: EL6695

EtherCATブリッジターミナル

EL6695 EtherCATブリッジターミナルは、EtherCATネットワークと他のマスタ間のリアルタイムデータ交換を可能にします。各種プロトコルによる非同期通信もサポートしています。ディストリビュートクロックは双方向で同期できます。EL6695はEL6692にはなかった、CoEコンフィグレーションの柔軟性、デバイスエミュレーションオプション、およびデータスループットの大幅な向上を実現しています (EL6692も引き続きお求めいただけます)。EL6692と同様、TwinCAT System Managerエクステンションには、便利なコンフィグレーションインターフェイスが用意されています。動作電源は、セカンダリ (RJ 45) には外部接続経由で供給し、プライマリにはEバス経由で供給します。柔軟なCoE構成により、このブリッジターミナルを使用して、従属的IPCシステムをEtherCATスレーブとして統合することもできます。この場合、ユーザはCoEで設定するパラメータを定義し、サブシステムを外部的にはユーザ定義のEtherCATスレーブとすることが可能です。

2.2 技術データ

技術データ	EL6695
ポート	プライマリ: Eバス(ターミナルブロック)、セカンダリ: 2 x 100 Mbit/s イーサネットRJ 45、入出力
機能	EtherCATディストリビュートクロック同期、同期/非同期データ交換
ケーブル長	セカンダリポート: 最長100 mの100BASE-TX
ハードウェア診断	ステータスLED
電源 消費電流	プライマリ: Eバス経由、定格400 mA セカンダリ: 24 V DC (-15%/+20%)、定格80 mA、プラグ着脱可能 動作には、片方の電圧のみ必要です。両方に電圧が加わっている場合、内部電源ユニットは優先的に24 V電源を使用します。
ディストリビュートクロック	あり、TC3.1 b4018.4以降のTwinCATが必要
電氣的絶縁	500 V (Eバス/セカンダリ)

技術データ	EL6695
周期プロセスデータ	各方向において最大 3 kbyte 注意: サポートする周期PDOの数は、主に使用するEcMasterによって異なります。 TC2. 11b2248またはTC3. 1 b4018の場合: 255の変数、最大全体サイズMTU (1400バイト未満)
サポートしている非同期プロトコル	CoE、EoE、AoE、FoE (VoE、SoE)
PDO送信速度	動作モードおよびデータ量によって異なります (通常は10~100 μs)。
最小EtherCATサイクルタイム	50 μs
特徴	<ul style="list-style-type: none"> ● TwinCAT内で外部リファレンスクロックとして使用可能 ● 同期データ交換 ● 柔軟なCoE定義/CoEデバイスエミュレーション ● 直接双方向DC同期として使用可能 ● PDO Mapping (対称または選択) ● ADS通信 (2つのマスタからのADS確認ダイアログを処理) ● FoEあたりの最大ユーザデータ送信速度5 kbyte
使用/保存周囲温度	0~+55 °C / -25~+85 °C
使用周囲湿度	95 %、結露なし
耐振性/耐衝撃性	EN 60068-2-6/EN 60068-2-27に準拠
EMCイミュニティ/エミッション	EN 61000-6-2/EN 61000-6-4に準拠
保護等級/設置位置	IP 20/可変
寸法	幅: 24 mm (並行設置) 高さ: 100 mm 奥行き: 68 mm
認証	CE cULus [▶ 33]

2.2.1 ベッコフEtherCATデータ交換デバイスの比較

	EL6695	EL6692	FC1100	FC1121	CXnnnn-B110
設計	24mmターミナル		PCIeプラグインカード		組込み型PCIに統合
PDO周期 プロセスデータ	各方向最大3 kB (1 TC2. 11b2248またはTC3. 1 b4018 実行時: 255の変数、 最大全体サイズMTU (約1400 バイト)	各方向最大480バイト	各方向最大1024バイト		各方向最大480バイト
PDO 送信速度 (デバイス内部、バス サイクル なし)	例: 入力200バイト/出力200 バイト: 定格15 μs 例: 入力1400バイト/出力 1400バイト: 定格50 μs	定格1~4 ms 例: 入力200バイト/出力 200バイト: 定格1 ms	例: 入力480バイト/出力480 バイト: 約300 μs (2)		例: 入力480バイト/出力 480バイト: 約250 μs、 CX5020以上 (2)
サポートしている 非同期 プロトコル	CoE、EoE、AoE、FoE (VoE、 SoE)	CoE、AoE、EoE			
メールボックス	128~1498バイト	128~1024バイト	64~1024バイト		
メールボックスの デフォルト設定	1024バイト	256バイト	1024バイト	512バイト	
最小許容 EtherCATサイクルタ イム(3)	50 μs (SyncMan割り込み使 用)	制限なし (SyncMan割り込み不使用)			
ディストリビュート クロック 同期	あり				
TwinCATで 外部リファレンスク ロックとして 使用可能	あり				

	EL6695	EL6692	FC1100	FC1121	CXnnnn-B110
固有のプロパティ	<ul style="list-style-type: none"> 同期データ交換 柔軟なCoE定義/CoEデバイスエミュレーション 直接双方向DC同期が可能 PDO Mapping (対称または選択) ADS通信 (2つのマスタによるADSリクエストを処理) FoEにより最大5 kBのユーザーデータを送信 独立したプライマリ/セカンダリ電源 	<ul style="list-style-type: none"> EC-マスタでの間接的なDC同期が可能 ADS通信 			

)¹ サポートしている周期PDOの数は、EtherCATマスタによって異なります。

)² 送信値は、IPC環境およびコントロールによって大幅に異なります。

)³ この制限は、デバイスの動作制限を意味します。両側の最小かつ妥当なサイクルタイムは、特に送信するデータ量（およびPDO送信時間）によって異なります。ターミナルが各サイクルでデータを取得/設定できるように、検出可能なサイクルタイムを選択する必要があります。

2.3 設置に関する動作条件

EL6695 EtherCATターミナルの輸送、保管、および動作は、必ず指定された条件下(技術データを参照)で行ってください。

警告

無効な動作条件での動作はおやめください

EL6695は、以下の動作条件では使用しないでください。

- 電離放射線の影響下
- 腐食環境下
- バスターミナルが許容範囲を超えて汚損される環境下

設置時の安全に関する指示事項

EtherCATターミナルの設置およびコミッショニングの前に、本取扱説明書の前書きに記載された安全に関する指示事項をお読みください。

輸送/保管

EtherCATターミナルの輸送または保管時には、必ず納入時の梱包資材をご使用ください。

電気配線

Eバスおよび24 V電源の両方が使用可能な場合、EL6695は優先的に24 V電源を使用することに注意してください。このため、EL6695がEバス経由で供給されるようにするためには、24 V電源が切断された状態でコミッショニングすることを推奨します。24 V電源が切断されていないと、Eバス電源の容量が誤っていた場合、24 V電源の故障時にターミナルネットワーク内のEバス電源で過負荷が発生する可能性があります。

Number	Box Name	Address	Type	In Size	Out Size	E-Bus (mA)
1	Box 1 (EL6695)	1001	EL6695	2.0		
2	Term 2 (EK1100)	1002	EK1100			
3	Term 3 (EL6695)	1003	EL6695	2.0		1600
4	Term 4 (EL1008)	1004	EL1008	1.0		1510
5	Term 5 (EL1008)	1005	EL1008	1.0		1420

3 基本的な通信

3.1 EtherCATの基本

EtherCATフィールドバスの基本については、『EtherCAT System Documentation』を参照してください。

3.2 EtherCATの配線 - 結線

2つのEtherCATデバイス間のケーブル長は、100 mを超えてはいけません。100 mというケーブル長は、特にケーブル長による信号減衰を考慮した場合、適切な特性のケーブル使用時の最大許容リンク長が5 + 90 + 5 mとなる高速イーサネットテクノロジーに由来します。『Design recommendations for the infrastructure for EtherCAT/Ethernet』も参照してください。

ケーブルおよびコネクタ

EtherCATデバイスの接続には、EN 50173またはISO/IEC 11801に準拠した、カテゴリ5 (Cat5)以上の要件を満たすイーサネット接続(ケーブル+プラグ)のみをご使用ください。EtherCATは、信号の送信に4本のワイヤを使用します。

EtherCATはRJ45プラグコネクタなどを使用します。ピン配置はイーサネット規格(ISO/IEC 8802-3)に対応しています。

ピン	コンダクタの色	信号	説明
1	黄色	TD +	送信データ +
2	オレンジ	TD -	送信データ -
3	白	RD +	受信データ +
6	青	RD -	受信データ -

自動ケーブル検出機能(auto-crossing)を搭載したベッコフのEtherCATデバイス間では、対称(1:1)またはクロスオーバーケーブルを使用できます。

i 推奨するケーブル

EtherCATデバイスの接続に適したケーブルは、ベッコフ ウェブサイトに記載されています。

Eバス電源

バスケーブルは接続されたELターミナルにEバスシステム電圧5 Vを供給できます。これにより、ケーブルは原則として最大2 Aまで供給可能です(詳細は該当するデバイスの取扱説明書を参照してください)。各ELターミナルのEバス消費電流値に関する情報は、ウェブおよびカタログに記載されています。接続されたターミナルの消費電流値の合計が、ケーブルの供給電流値を超過する場合は、Eバス電源供給ターミナル(EL9410など)をターミナルブロック内の適切な個所に挿入する必要があります。

設計値として理論上の最大Eバス電流は、TwinCAT System Managerに列の値として表示されます。電流不足は負の合計量とエクスクラメーションマークで示されます。電流不足が発生する個所の前に、電源ターミナルを追加する必要があります。

Number	Box Name	Add...	Type	In Si...	Out ...	E-Bus (mA)
1	Term 1 (EK1100)	1001	EK1100			
2	Term 2 (EL2008)	1002	EL2008		1.0	1890
3	Term 3 (EL2008)	1003	EL2008		1.0	1780
4	Term 4 (EL2008)	1004	EL2008		1.0	1670
5	Term 5 (EL6740...)	1005	EL6740-0010	2.0	2.0	1220
6	Term 6 (EL6740...)	1006	EL6740-0010	2.0	2.0	770
7	Term 7 (EL6740...)	1007	EL6740-0010	2.0	2.0	320
8	Term 8 (EL6740...)	1008	EL6740-0010	2.0	2.0	-130 !
9	Term 9 (EL6740...)	1009	EL6740-0010	2.0	2.0	-580 !

図 11: System Managerでの電流計算

注記

誤作動の可能性あり

ターミナルブロック内のすべてのEtherCATターミナルのEバス電源に対して、同一の接地電位を使用する必要があります。

3.3 ウォッチドッグ設定に関する一般的な注記

ELxxxターミナルには、プロセスデータ通信の中断時など、指定した時間の経過後にデバイスおよび設定(OFF状態など)に応じて出力をオフに切り替えるセーフティ機能(ウォッチドッグ)が搭載されています。

EL2xxxターミナル内のEtherCATスレーブコントローラ(ESC)には、以下の2つのウォッチドッグが用意されています。

- ・ SMウォッチドッグ(デフォルト: 100 ms)
- ・ PDIウォッチドッグ(デフォルト: 100 ms)

SMウォッチドッグ(SyncManagerウォッチドッグ)

SyncManagerウォッチドッグは、ターミナルとのEtherCATプロセスデータ通信が成功するたびにリセットされます。ラインの切断時など、設定かつ有効化したSMウォッチドッグ時間が経過してもターミナルとのEtherCATプロセスデータ通信が行われない場合は、ウォッチドッグがトリガされ、出力がFALSEにセットされます。ターミナルのOP状態は変化しません。ウォッチドッグは、EtherCATプロセスデータアクセスに成功しないとリセットされません。以下の説明にしたがって、モニタリング時間を設定します。

SyncManagerウォッチドッグは、EtherCAT側からESCとの正常かつ正確なタイミングでプロセスデータ通信が行われているかを監視します。

PDIウォッチドッグ(プロセスデータウォッチドッグ)

設定かつ有効化したPDIウォッチドッグ時間が経過してもEtherCATスレーブコントローラ(ESC)とのPDI通信が行われない場合は、このウォッチドッグがトリガされます。

PDI(プロセスデータインターフェイス)は、ESCとEtherCATスレーブ内のローカルプロセッサなどの内部インターフェイスです。PDIウォッチドッグを使用すると、この通信が失敗していないかをモニタリングできます。

PDIウォッチドッグは、アプリケーション側からESCとの正常かつ正確なタイミングでプロセスデータ通信が行われているかを監視します。

SMウォッチドッグおよびPDIウォッチドッグの設定は、TwinCAT System Managerで各スレーブに対して個別に行う必要があります。

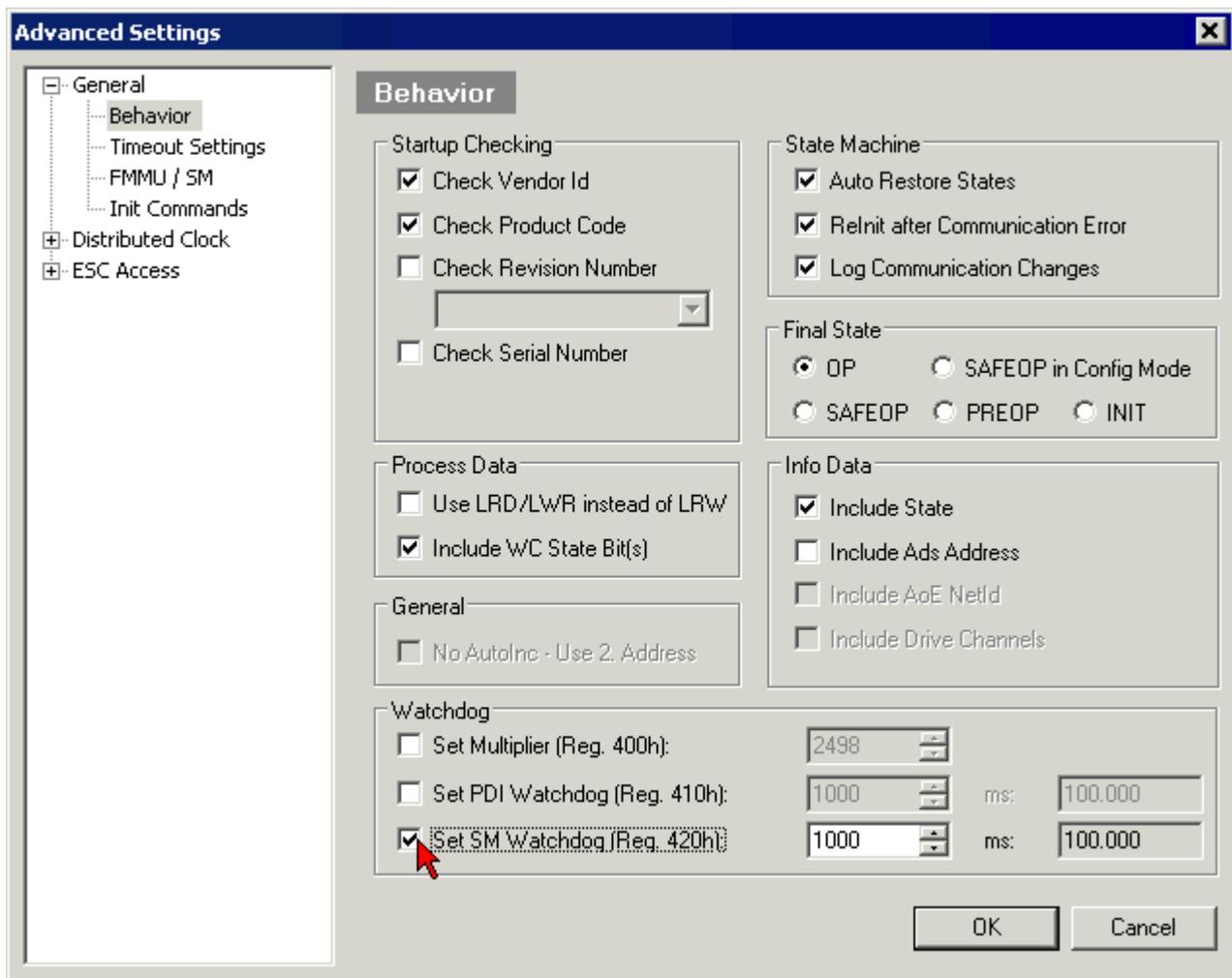


図 12: [EtherCAT]タブ -> [Advanced Settings] -> [Behavior] -> [Watchdog]

注記:

- ・ 乗算は両方のウォッチドッグに対して有効です。
- ・ 各ウォッチドッグには独自のタイマ設定が用意されています。乗数を掛けた結果が設定時間となります。
- ・ 重要: 乗数/タイマ設定は、チェックボックスが有効な場合にのみ、スタートアップ時にスレーブにロードされます。チェックボックスが無効な場合は、何もダウンロードされず、ESC設定は変更されません。

乗数

乗数

どちらのウォッチドッグも、ウォッチドッグ乗数によって除算された、ローカルターミナルサイクルからパルスを受信します。

$$1/25 \text{ MHz} * (\text{ウォッチドッグ乗数} + 2) = 100 \text{ } \mu\text{s} \text{ (乗数のデフォルト設定2498)}$$

SMウォッチドッグの標準設定1000は、100 msの解放時間と一致します。

乗数の値 + 2は、ウォッチドッグの1回のティックを示す基本の40 nsティックの数と一致します。乗数を変更し、ウォッチドッグ時間の設定範囲が大きくなるように調整できます。

[Set SM watchdog]の例

このチェックボックスにより、ウォッチドッグ時間の手動設定が可能になります。出力が設定されていて、EtherCAT通信が中断されると、設定した時間の経過後にSMウォッチドッグがトリガされ、出力が消去されず。この設定を使用して、より低速なEtherCATマスタ、または長いサイクルタイムをターミナルに適合することが可能です。デフォルトのSMウォッチドッグ設定は100 msです。設定範囲は0~65535です。乗数と1~65535の範囲を組み合わせることで、0~170秒のウォッチドッグ時間をカバーできます。

計算

乗数 = 2498 → ウォッチドッグ基本時間 = $1 / 25 \text{ MHz} * (2498 + 2) = 0.0001 \text{ 秒} = 100 \text{ } \mu\text{s}$
 SMウォッチドッグ = 10000 → $10000 * 100 \text{ } \mu\text{s} = 1 \text{ 秒}$ のウォッチドッグモニタリング時間

⚠ 注意

未定義の状態となる可能性があります。

SMウォッチドッグ = 0とすることでSMウォッチドッグをオフに切り替える機能は、バージョン-0016以降のターミナルにのみ実装されています。これ以前のバージョンでは、この動作モードは使用してはいけません。

⚠ 注意

デバイスが損傷し、未定義の状態となる可能性があります。

SMウォッチドッグが有効な場合、値0を入力すると、ウォッチドッグが完全にオフに切り替わります。これにより、ウォッチドッグが無効になります。通信が中断している場合でも出力が安全な状態に設定されません。

3.4 EtherCATステートマシン

EtherCATスレーブの状態は、EtherCATステートマシン (ESM) によって制御されます。状態に応じて、EtherCATスレーブ内で異なるファンクションへのアクセスおよび実行が可能になります。特にスレーブの起動中は、各状態で特定のコマンドをEtherCATマスタがデバイスに送信する必要があります。

以下の状態が区別されます。

- ・ Init
- ・ Pre-Operational
- ・ Safe-Operational
- ・ Operational
- ・ Boot

起動後の各EtherCATスレーブの通常の状態は、OP状態です。

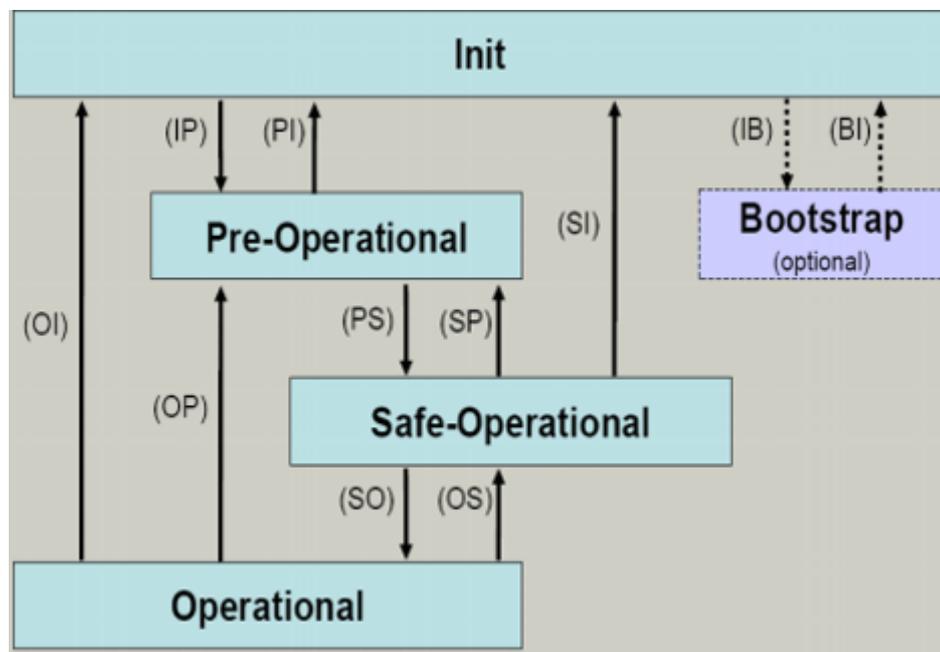


図 13: EtherCATステートマシンの状態

Init

EtherCATスレーブのスイッチをオンにすると、Init状態となります。メールボックス通信またはプロセスデータ通信はできません。EtherCATマスタは、メールボックス通信用にSync Managerチャンネル0および1を初期化します。

Pre-Operational (Pre-Op)

*Init*から*Pre-Op*への遷移中、EtherCATスレーブはメールボックスが正常に初期化されたかどうかをチェックします。

*Pre-Op*状態では、メールボックス通信は可能ですが、プロセスデータ通信はできません。EtherCATマスタは、プロセスデータのSync Managerチャンネル(Sync Managerチャンネル2から)、FMMUチャンネル、およびスレーブが構成可能なマッピングをサポートしている場合はPDO Mapping、またはSync Manager PDO割り当てを初期化します。この状態では、プロセスデータ通信の設定、およびデフォルト設定から変更するターミナル固有のパラメータも送信します。

Safe-Operational (Safe-Op)

*Pre-Op*から*Safe-Op*への遷移中、EtherCATスレーブはプロセスデータ通信用のSync Managerチャンネルをチェックし、必要に応じてディストリビュートクロック設定を行います。EtherCATスレーブは、状態の変化を確認する前に、現在の入力データをEtherCATスレーブコントローラ(EGSC)の関連するDP-RAM領域にコピーします。

*Safe-Op*状態では、スレーブは出力を安全な状態に保ち、入力データを周期的に更新しますが、メールボックス通信およびプロセスデータ通信は可能です。

● SAFEOP状態の出力

I デフォルト設定でのウォッチドッグ [▶ 17]モニタリングは、設定がSAFEOPであるかOP (OFF状態など)であるかに応じて、モジュールの出力を安全な状態に設定します。モジュール内でウォッチドッグモニタリングが無効になっていて、これが行われない場合は、出力もSAFEOP状態に切り替わるか、SAFEOP状態に設定される可能性があります。

Operational (Op)

EtherCATマスタは、EtherCATスレーブを*Safe-Op*から*Op*に切り替える前に、有効な出力データを送信する必要があります。

*Op*状態では、スレーブはマスタの出力データを自身の出力にコピーします。プロセスデータ通信およびメールボックス通信は可能です。

Boot

*Boot*状態では、スレーブのファームウェアを更新できます。*Init*状態からのみ、*Boot*に移行できます。

*Boot*状態では、*File access over EtherCAT* (FoE) プロトコル経由でのメールボックス通信は可能ですが、その他のメールボックス通信およびプロセスデータ通信はできません。

3.5 CoEインターフェイス

概要説明

CoE (CANopen over EtherCAT) インターフェイスは、EtherCATデバイスのパラメータ管理に使用します。EtherCATスレーブやEtherCATマスタは、操作、診断、またはコミッショニングに必要な固定(読み取り専用)または可変パラメータを管理します。

CoEパラメータは、テーブル階層に配置されます。原則として、ユーザはフィールドバス経由での読み取りアクセスが可能です。EtherCATマスタ(TwinCAT System Manager)は、属性に応じて、読み取りまたは書き込みモードでスレーブのローカルCoEリストへのEtherCAT経由でのアクセスが可能です。

文字列(テキスト)、整数値、Boolean値、より大きなバイトフィールドなど、さまざまなCoEパラメータタイプを使用でき、各種機能を参照、設定できます。これらのパラメータの例として、メーカID、シリアル番号、プロセスデータ設定、デバイス名、アナログ計測の補正值、パスワードなどが挙げられます。

順序は16進数のナンバリングによって、(メイン)インデックスとそれに続くサブインデックスの2つのレベルで指定されます。値の範囲は以下のとおりです。

- ・ インデックス: 0x0000~0xFFFF (0~65535_{dez})
- ・ サブインデックス: 0x00~0xFF (0~255_{dez})

通常、この方法でローカライズされたパラメータは0x8010:07のように表記されます。先頭部分の「x」が16進数値範囲を識別し、コロンのインデックスとサブインデックスを区切ります。

EtherCATフィールドバスユーザに関連する範囲は、以下のとおりです。

- ・ 0x1000: ここには、名前、メーカー、シリアル番号などを含むデバイスの固定識別情報、および最新かつ使用可能なプロセスデータ設定に関する情報が格納されます。
- ・ 0x8000: ここには、フィルタ設定や出力周波数などの、すべてのチャンネルの動作および機能に関するパラメータが格納されます。

その他の重要な範囲は、以下のとおりです。

- ・ 0x4000: EtherCATデバイスによっては、ここに(0x8000範囲の代替として)チャンネルパラメータが格納されます。
- ・ 0x6000: 入力PDO (EtherCATマスタ側からの「入力」)
- ・ 0x7000: 出力PDO (EtherCATマスタ側からの「出力」)

● 可用性

i CoEリストをもたないEtherCATデバイスもあります。通常、専用プロセッサを搭載していない単純なI/Oモジュールには可変パラメータがないため、CoEリストもありません。

デバイスにCoEリストがある場合は、TwinCAT System Manager内の個別のタブにエレメントのリストが表示されます。

Index	Name	Flags	Value
1000	Device type	RO	0x00FA1389 (16389001)
1008	Device name	RO	EL2502-0000
1009	Hardware version	RO	
100A	Software version	RO	
1011:0	Restore default parameters	RO	> 1 <
1018:0	Identity	RO	> 4 <
1018:01	Vendor ID	RO	0x00000002 (2)
1018:02	Product code	RO	0x09C63052 (163983442)
1018:03	Revision	RO	0x00130000 (1245184)
1018:04	Serial number	RO	0x00000000 (0)
10F0:0	Backup parameter handling	RO	> 1 <
1400:0	PWM RxDPO-Par Ch.1	RO	> 6 <
1401:0	PWM RxDPO-Par Ch.2	RO	> 6 <
1402:0	PWM RxDPO-Par h.1 Ch.1	RO	> 6 <
1403:0	PWM RxDPO-Par h.1 Ch.2	RO	> 6 <
1600:0	PWM RxDPO-Map Ch.1	RO	> 1 <

図 14: [CoE Online]タブ

上図は、0x1000~0x1600のデバイス「EL2502」で使用可能なCoEオブジェクトを示しています。0x1018のサブインデックスは展開されています。

データ管理および「NoCoeStorage」機能

特にスレーブの設定パラメータなど、設定および書き込みが可能なパラメータもあります。これは、以下の方法で書き込みモード、または読み取りモードで行えます。

- ・ System Manager (図. [CoE Online]タブ)をクリック
これは、システム/スレーブのコミッショニング時に便利です。パラメータ設定するインデックスの行をクリックし、[SetValue]ダイアログで値を入力します。
- ・ TcEtherCAT.libライブラリのブロックなどからADS経由で制御システム/PLCを使用
これは、システムの動作中、またはSystem Managerが使用できない場合や運用スタッフの不在時に変更を行う場合に推奨されます。

● データ管理

スレーブのCoEパラメータがオンラインを変更すると、ベッコフ デバイスはフェールセーフな方法でEEPROM内にあらゆる変更を格納します。これにより、変更したCoEパラメータは、再起動後も使用可能な状態で維持されます。
この動作は、他のメーカーとは異なる可能性があります。

書き込み動作に関しては、EEPROMのライフタイムには限度があります。通常、100,000回の書き込み動作以降は、新しい(変更した)データが確実に保存される、または読み取れるという保証がありません。これは、通常のコミッショニングでは考慮する必要はありません。ただし、マシンのランタイム時にCoEパラメータをADS経由で継続的に変更する場合は、ライフタイムの限度に達する可能性が大いに考えられます。変更したCoE値の保存を抑制するNoCoeStorage機能をサポートしているかどうかは、ファームウェアバージョンによって異なります。
デバイスでこの機能をサポートしているかどうかは、本取扱説明書の技術データでご確認ください。

- ・ この機能をサポートしている場合: CoE 0xF008にコードワード0x12345678を入力するとこの機能が有効になり、コードワードが変更されなければ有効な状態で維持されます。デバイスをオンに切り替えると、この機能は無効になります。変更したCoE値がEEPROMに保存されなくなるため、何回でも変更できます。
- ・ この機能がサポートされていない場合: ライフタイムの限度を考慮し、CoE値の継続的な変更は許可されません。

● スタートアップリスト

ターミナルを交換すると、ターミナルのローカルCoEリスト内の変更は消失します。ターミナルを新しいベッコフ ターミナルと交換すると、デフォルト設定となります。このため、EtherCATフィールドバスを開始すると必ず処理されるスレーブのスタートアップリストによって、EtherCATスレーブのCoEリスト内すべての変更をリンクすることを推奨します。この方法により、交換するEtherCATスレーブをユーザの指定によって自動的にパラメータ設定できます。

ローカルCoE値を継続的に保存できないEtherCATスレーブを使用する場合は、スタートアップリストを使用する必要があります。

推奨するCoEパラメータの手動での変更方法

- ・ System Manager内で必要な変更を行います。
値がEtherCATスレーブ内でローカルに保存されます。
- ・ 値を継続的に保存する場合は、値をスタートアップリストに入力します。
通常、スタートアップエントリの順序は関係ありません。

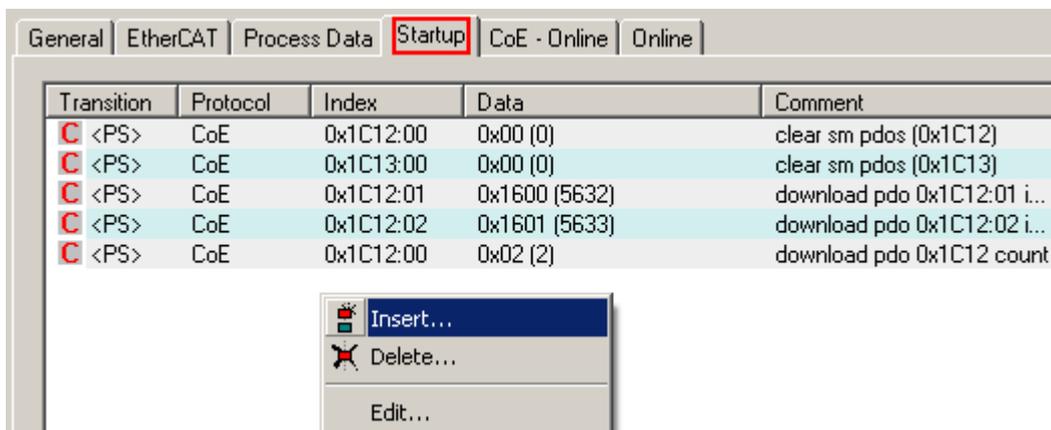


図 15: TwinCAT System Managerに表示されたスタートアップリスト

スタートアップリストには、ESI仕様に基づいてSystem Managerが設定した値が既に含まれている場合があります。アプリケーション固有の追加エントリを作成できます。

オンライン/オフラインリスト

TwinCAT System Managerでの作業時には、EtherCATデバイスが「使用可能」であるか(スイッチがオンかつEtherCAT経由で接続されており、オンラインである)、またはスレーブが接続されていないオフライン状態で設定が作成されたかを区別する必要があります。

どちらの場合も、図. [CoE online]タブのようなCoEリストが表示されます。接続されているかどうか、オンライン/オフラインとして示されます。

- ・ スレーブがオフラインの場合

- ESIファイルのオフラインリストが表示されます。この場合、変更ができないか、変更しても効果がありません。
- 設定したステータスは、[Identity]に表示されます。
- ファームウェアやハードウェアバージョンは物理デバイスの属性であるため、表示されません。
- 赤で**Offline**と表示されます。

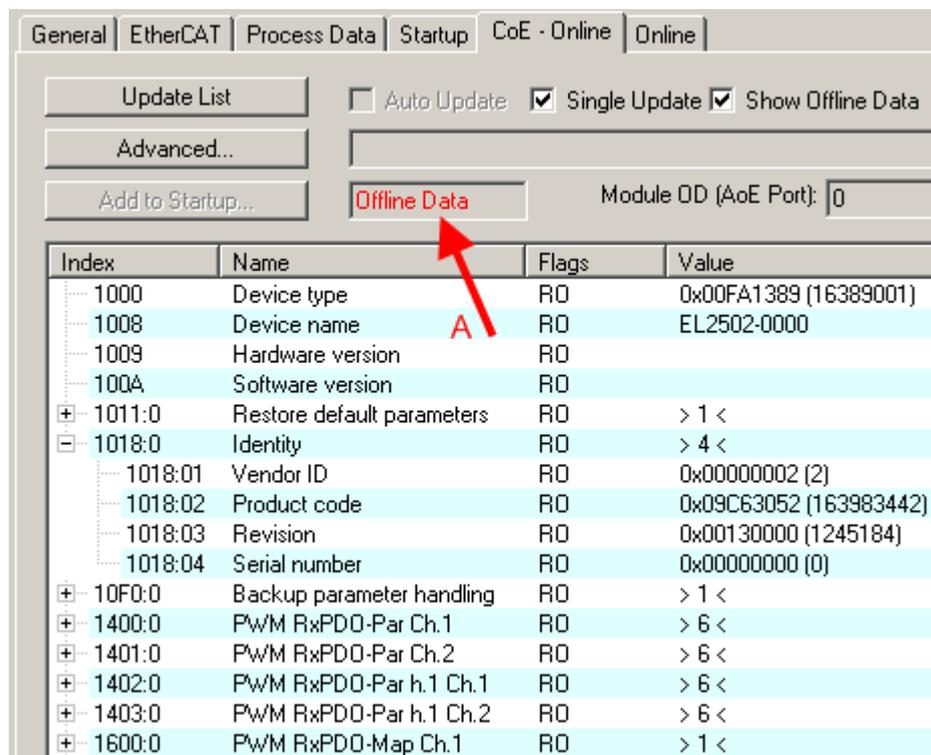


図 16: オフラインリスト

- ・ スレーブがオンラインの場合
 - 実際の現在のスレーブリストが読み取られます。サイズおよびサイクルタイムによっては、読み取りに数秒かかることがあります。
 - 実際の識別情報が表示されます。
 - 電子情報に基づいて、機器のファームウェアおよびハードウェアバージョンが表示されます。
 - 緑で**Online**と表示されます。

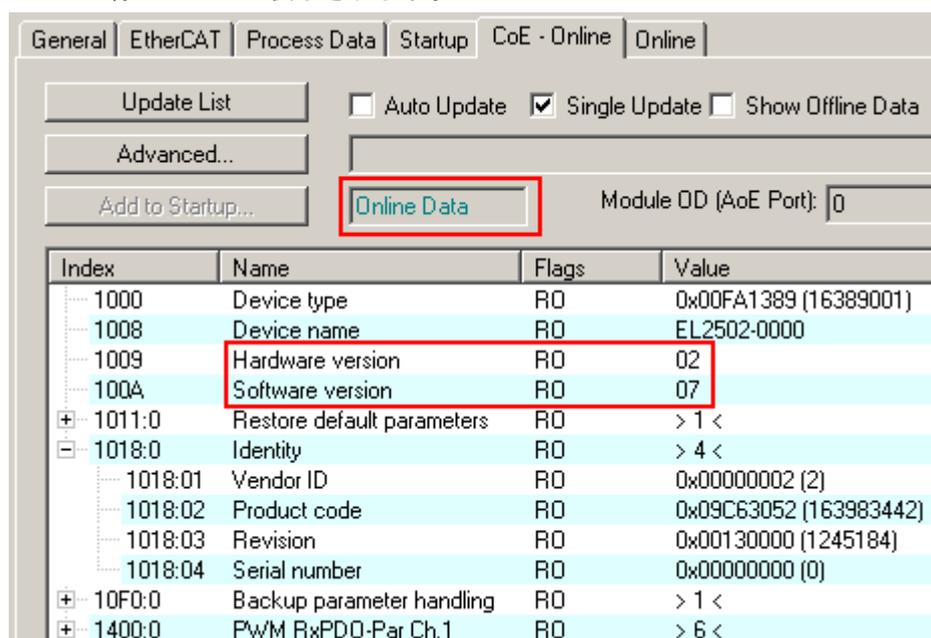


図 17: オンラインリスト

チャンネルベースのオーダー

通常、EtherCATデバイスのCoEリストには、複数の機能的に同等なチャンネルが用意されています。例えば、4チャンネルアナログ0~10 V入力ターミナルには4つの論理チャンネルも用意されているため、チャンネルに対して4つの同一なパラメータデータのセットが存在することになります。本取扱説明書では、各チャンネルについて列記することを避けるため、個々のチャンネル番号に対してプレースホルダ「n」を使用します。

通常、CoEシステムでは、それぞれ255個のサブインデックスをもつ16個のインデックスがあればすべてのチャンネルパラメータを表現できます。このため、チャンネルベースのオーダーは $16_{\text{dec}}/10_{\text{hex}}$ ステップに配置されます。例として、パラメータ範囲0x8000では以下ようになります。

- ・ チャンネル0: パラメータ範囲0x8000:00~0x800F:255
- ・ チャンネル1: パラメータ範囲0x8010:00~0x801F:255
- ・ チャンネル2: パラメータ範囲0x8020:00~0x802F:255
- ・ ...

通常、これは0x80n0と記述されます。

CoEインターフェイスに関する詳細情報は、ベッコフ ウェブサイトの [EtherCATシステムマニュアル](#)に記載されています。

3.6 ディストリビュートクロック

EtherCATスレーブコントローラ(ESC)内のローカルクロックを示すディストリビュートクロックには、以下の特性があります。

- ・ 単位 1 ns
- ・ 原点 $1.1.2000\ 00:00$
- ・ サイズ 64 ビット (以降の584年間に対応。ただしEtherCATスレーブによっては32ビットしかサポートしないため、変数が約4.2秒後にオーバーフローします。)
- ・ EtherCATマスタは、ローカルクロックとEtherCATバス内のマスタクロックを誤差100 ns未満の精度で同期します。

詳細情報は、[EtherCATシステムの説明](#)を参照してください。

4 取付けおよび配線

4.1 ESD保護に関する指示事項

注記

静電気放電によるデバイス破損の危険

このデバイスには、不適切な取り扱いによって生じる静電気放電の影響を受けるコンポーネントが含まれています。

- ・ 静電気放電されていることを確認し、デバイスの接点に直接触れないようにしてください。
- ・ 絶縁性の高い物質(合成繊維、プラスチックフィルムなど)への接触は避けてください。
- ・ デバイスを扱う際には、周囲環境(作業場所、梱包材、および作業員)が適切に接地されている必要があります。
- ・ 保護クラスおよびESD保護を確保するために、各アセンブリの右側の終端をEL9011またはEL9012バスエンドキャップで保護する必要があります。

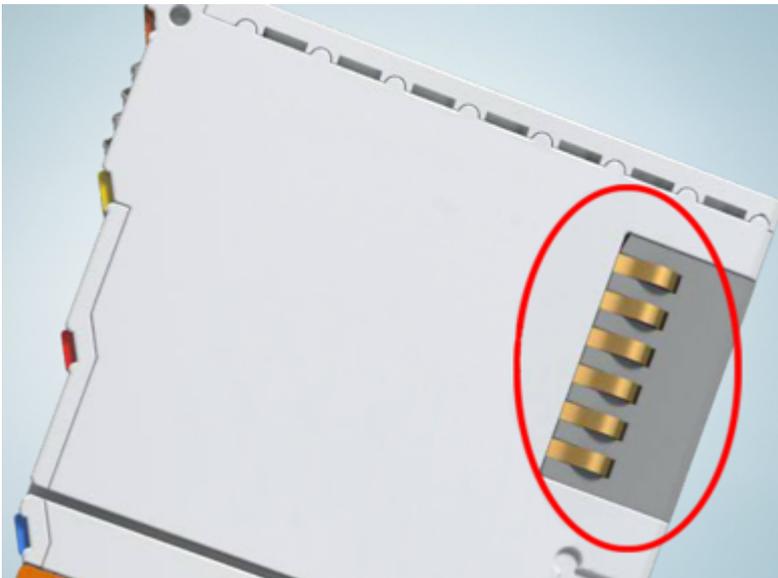


図 18: ベッコフI/O機器のデータ通信用端子

4.2 取付けおよび取外し - フロントロック解除式ターミナル

ターミナルモジュールは、35 mm取付けレール(取付けレールTH 35-15など)の形状により、取付け面に固定することができます。

● 取付けレールの固定

i ターミナルおよびカプラのロック機構は、取付けレールの背面まで到達します。取付け時に、コンポーネントのロック機構が取付けレールの固定ボルトに干渉しないようにしてください。推奨する取付けレールをターミナルおよびカプラの下に取り付けるには、フラットな取付け金具(さらネジやブラインドリベットなど)を使用する必要があります。

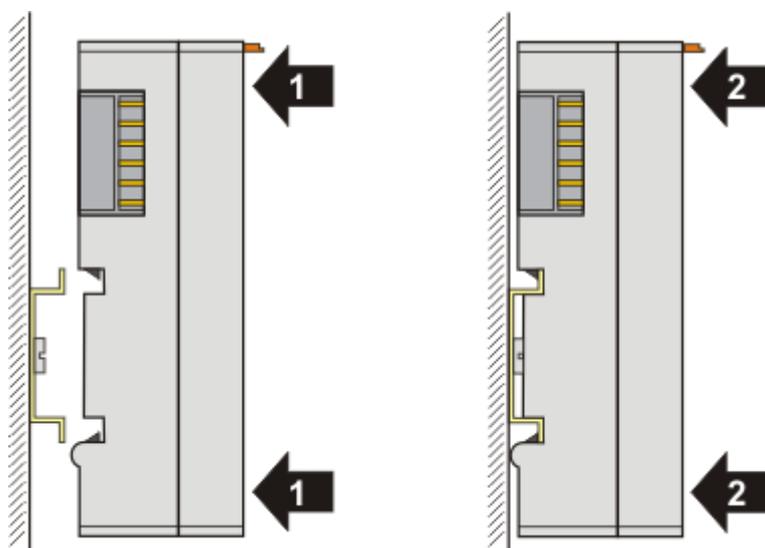
⚠ 警告

感電およびデバイスの損傷のリスク

バスターミナルの設置、取外し、または配線の前に、バスターミナルシステムを安全かつ通電していない状態にしてください。

取付け

- ・ 取付けレールを目的の取付け位置に固定します。

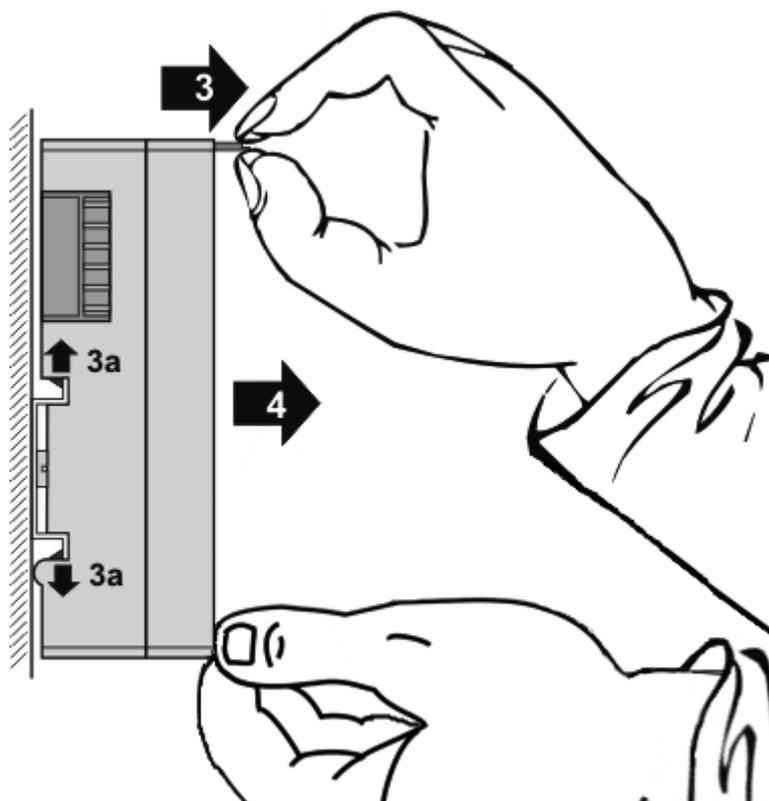


ターミナルモジュールがカチッとハマるまで、取付けレールに押し付けます(1)(2)。

- ・ ケーブルを取り付けます。

取外し

- ・ ケーブルをすべて取り外します。
- ・ 親指と人差し指で、取外しフックを引き出します。内部機構が2つの取付けラグ(3a)をレールのつめからターミナルモジュールに引き入れ、



- ・ ターミナルモジュールを取付け面から引き外します(4)。モジュールは傾かないようにしてください。必要に応じて、もう一方の手でモジュールを押さえてください。

4.3 推奨する取付けレール

KMxxxxおよびEMxxxxシリーズのターミナルモジュールおよびEtherCATモジュールは、EL66xxおよびEL67xxシリーズのターミナル同様、推奨する以下の取付けレールに直接、取り付けできます。

板厚1 mmのDINレールTH 35-7.5 (EN 60715準拠)

板厚1.5 mmのDINレールTH 35-15

● **DINレールの板厚に注意してください**



KMxxxxおよびEMxxxxシリーズのターミナルモジュールおよびEtherCATモジュールは、EL66xxおよびEL67xxシリーズのターミナル同様、板厚2.2~2.5 mmのDINレールTH 35-15 (EN 60715準拠)には適合しません。

4.4 設置方向

注記

設置方向および使用周囲温度範囲に関する制約

設置方向、使用周囲温度範囲、またはその両方に関する制約が定められていないか、ターミナルの技術データで確認してください。放熱量の大きなターミナルを設置する際には、ターミナルの上下の他のコンポーネントとの間に十分な隙間を開け、十分に換気を行うようにしてください。

最適な設置方向(標準)

設置方向を最適にするには、取付けレールを水平に設置し、EL/KLターミナルの配線部分が前面になるように設置する必要があります(図.「標準設置方向の推奨距離」)。ターミナルは下部から換気され、対流によって電子部品が最適に冷却されます。「下部から」換気されるのは、重力が作用するためです。

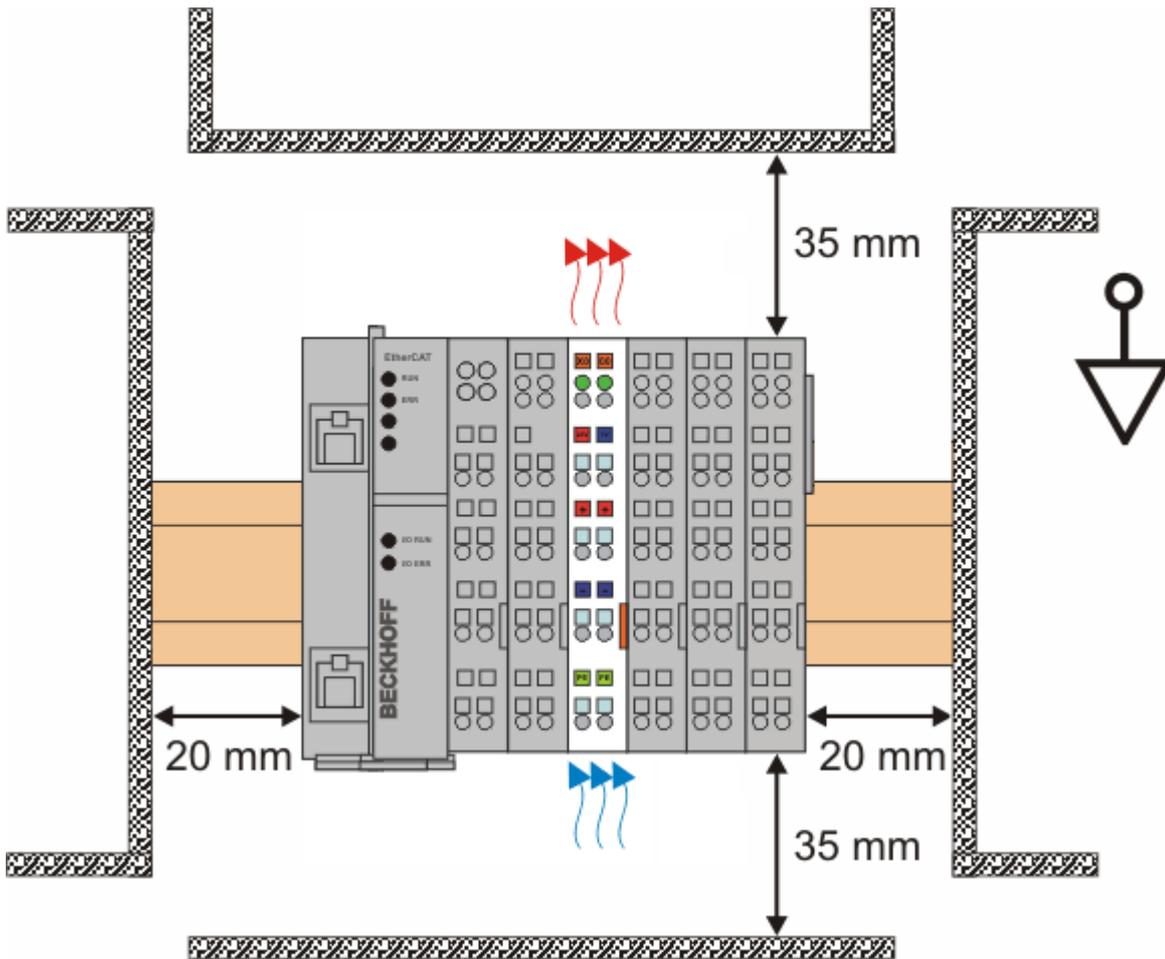


図 19: 標準設置方向の推奨距離

図.「標準設置方向の推奨距離」に記載されている距離を遵守することを推奨します。

その他の設置方向

その他の設置方向は、すべて取付けレールの設置方法によって決まります。図.「その他の設置方向」を参照してください。

上記の周辺との最小距離が、その他の設置方向にも適用されます。

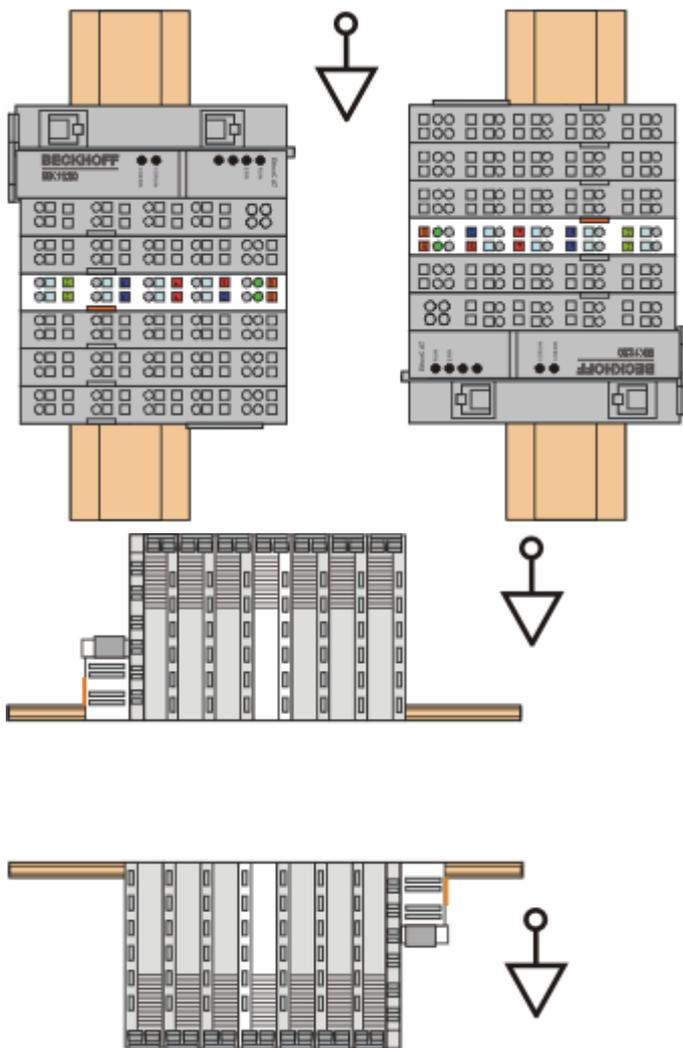


図 20: その他の設置方向

4.5 パッシブターミナルの配置

● バスターミナルブロック内でのパッシブターミナルの配置のポイント

i バスターミナルブロック内のデータ通信において自らデータのやりとりを行わないEtherCATターミナル(ELxxxx / ESxxxx)をパッシブターミナルと呼びます。パッシブターミナルは、Eバスからの電流を消費しません。
データ通信を適切に行うために、3つ以上のパッシブターミナルをつないで使用してはいけません。

パッシブターミナルの配置例(ハイライト部分)

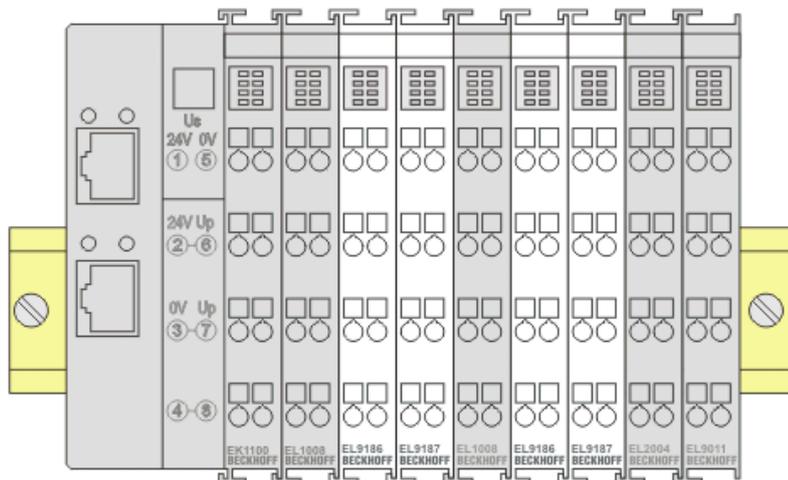


図 21: 正しい配置

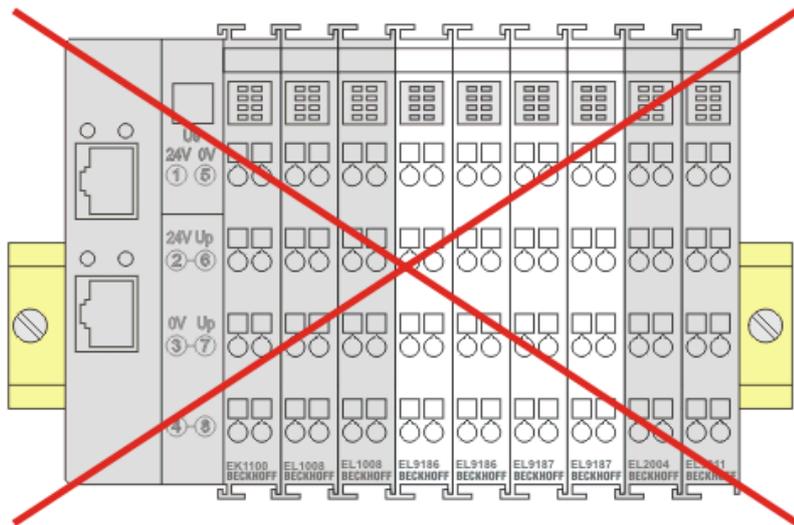


図 22: 間違った配置

4.6 ULに関する注記

	<p>用途 ベッコフEtherCATモジュールは、UL規格に適合したベッコフのEtherCATシステム専用です。</p>
	<p>試験 cULus試験では、ベッコフI/Oシステムは火災および感電のリスクについてのみ調査が行われています (UL508およびCSA C22.2 No. 142に準拠)。</p>
	<p>イーサネットコネクタ付きのデバイスについて 通信回線への接続用ではありません。</p>

基本原則

UL認証はUL508に準拠したものです。この種類の認証を受けたデバイスには、以下の記号が印字されています。



4.7 接続

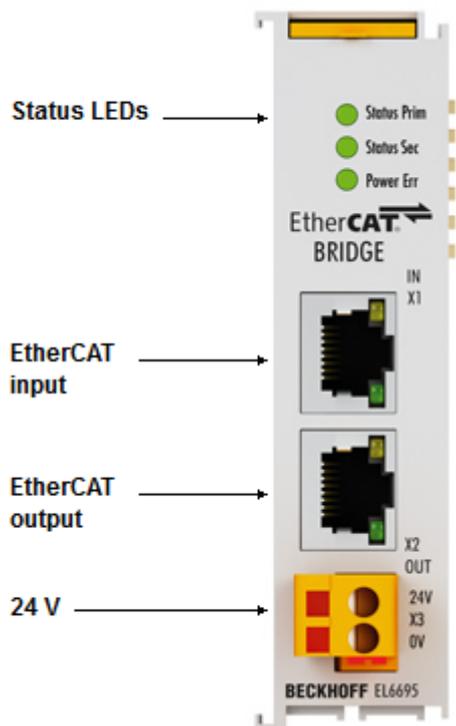


図 23: EL6695の接続

端子	説明
X1	EtherCAT入力 (RJ45、10BASE-T/100BASE-TX イーサネット)
X2	EtherCAT出力 (RJ45、10BASE-T/100BASE-TX イーサネット)
X3	2極ソケットターミナル接続 (24 VDC)、セカンダリ電源



LED診断

LEDに関する説明は、チャプタ「[LED診断 \[▶ 162\]](#)」を参照してください。

5 コミッショニング

5.1 TwinCATクイックスタート

TwinCATは、マルチPLCシステム、NC軸制御、プログラミング、およびオペレーションを含むリアルタイム制御用の開発環境です。この環境によってシステム全体がマッピングされ、コントローラのプログラミング環境(コンパイルを含む)にアクセスできます。機能の確認などの目的で、個別のデジタルまたはアナログ入出力の直接読み取り、および書き込みが可能です。

詳細情報は、<http://infosys.beckhoff.com>を参照してください。

- ・ **EtherCATシステムマニュアル:**
Fieldbus Components → EtherCAT Terminals → EtherCAT System Documentation → Setup in the TwinCAT System Manager
- ・ **TwinCAT 2** → TwinCAT System Manager → I/O - Configuration
- ・ TwinCATドライバのインストール:
Fieldbus components → Fieldbus Cards and Switches → FC900x - PCI Cards for Ethernet → Installation

デバイスには、実際のコンフィグレーション用のターミナルが含まれています。コンフィグレーションデータは、すべてエディタ機能(オフライン)または、スキャン機能(オンライン)経由で直接入力できます。

- ・ 「**オフライン**」: コンポーネントを個別に追加および配置することで、構成をカスタマイズできます。コンポーネントはツリーリストから選択して構成することが可能です。
 - オフラインモードでの手順は、<http://infosys.beckhoff.com>に記載されています。
TwinCAT 2 → TwinCAT System Manager → IO - Configuration → Adding an I/O Device
- ・ 「**オンライン**」: 接続したハードウェアコンフィグレーションが読み込まれます。
 - <http://infosys.beckhoff.com>も参照してください。
Fieldbus components → Fieldbus cards and switches → FC900x - PCI Cards for Ethernet → Installation → Searching for devices

ユーザPCから個々の制御エレメントまでには、以下のような関係が想定されます。

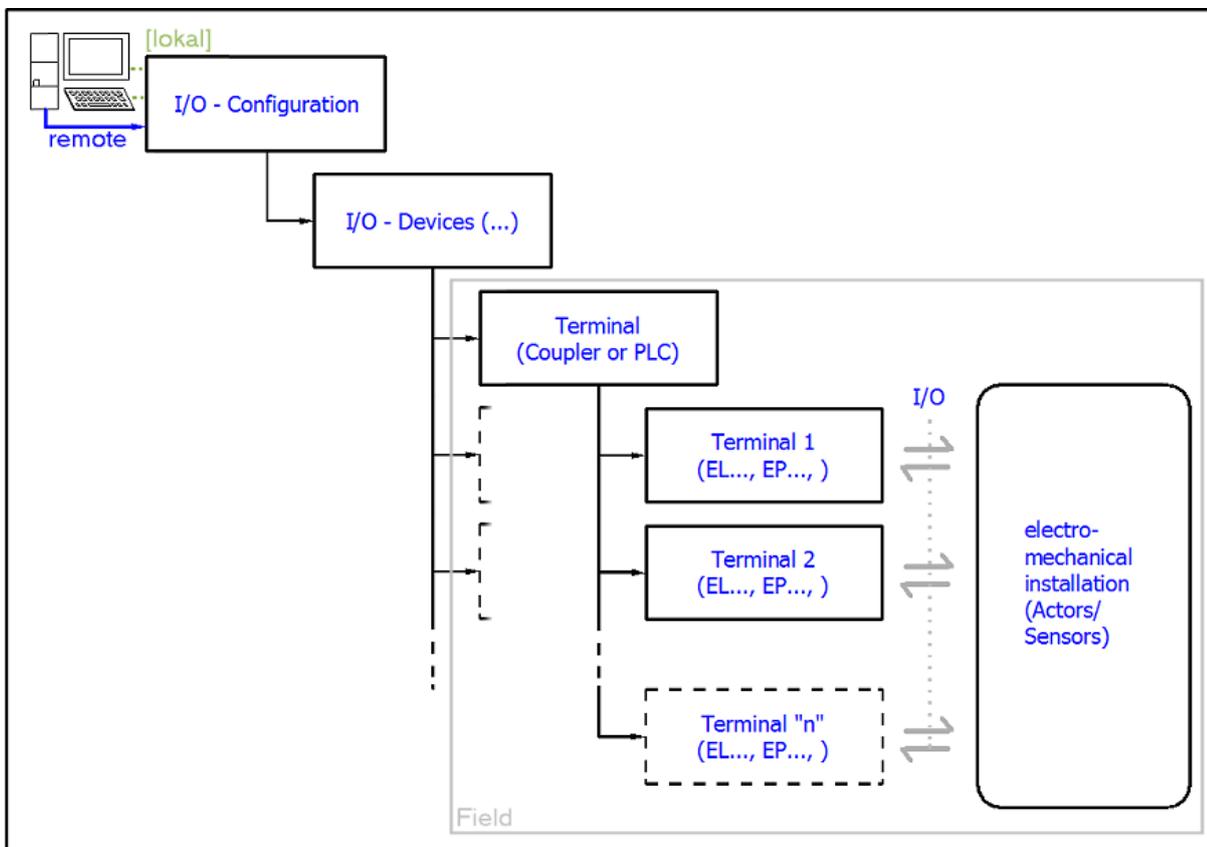


図 24: ユーザ側(コミッショニング)と設置の関係

特定のコンポーネント (I/Oデバイス、ターミナル、ボックスなど) を挿入するユーザは、TwinCAT 2でもTwinCAT 3でも同じです。以下は、オンラインでの手順に関する説明です。

コンフィグレーション例 (実際のコンフィグレーション)

以降のサブセクションでは、以下のコンフィグレーション例に基づいて、TwinCAT 2およびTwinCAT 3での手順を説明します。

- ・ CX2100-0004電源ユニットを含む制御システム (PLC) CX2040
- ・ CX2040の右側に接続 (Eバス) :
EL1004 (4チャンネルデジタル入力ターミナル24 V DC)
- ・ X001ポート (RJ-45) 経由で接続: EK1100 EtherCATカプラ
- ・ EK1100 EtherCATカプラの右側に接続 (Eバス) :
EL2008 (8チャンネルデジタル出力ターミナル24 V DC、0.5 A)
- ・ (X000経由 (オプション) : ユーザインターフェイスの外部PCへのリンク)

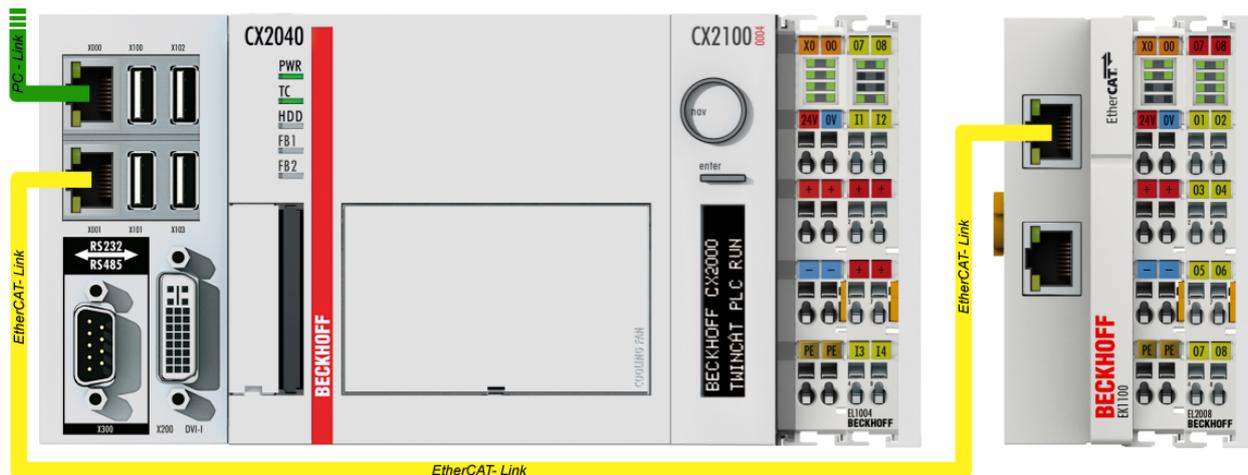


図 25: 組み込み型PCを使用した制御コンフィグレーション、入力 (EL1004) および出力 (EL2008)

すべての構成の組み合わせが可能です。例えば、EL1004ターミナルをカプラの後に接続することも、EL2008ターミナルをCX2040の右側に追加接続することも可能です。この場合、EK1100カプラは不要になります。

5.1.1 TwinCAT 2

スタートアップ

基本的に、TwinCATでは次の2つのユーザインターフェイスを使用します。電気機械的コンポーネントとの通信のTwinCAT System Managerと、コントローラの開発およびコンパイル用のTwinCAT PLC Controlです。TwinCAT System Managerから始めます。

開発に使用するPCにTwinCATシステムを正常にインストールすると、スタートアップ後にTwinCAT 2 System Managerに以下のユーザインターフェイスが表示されます。

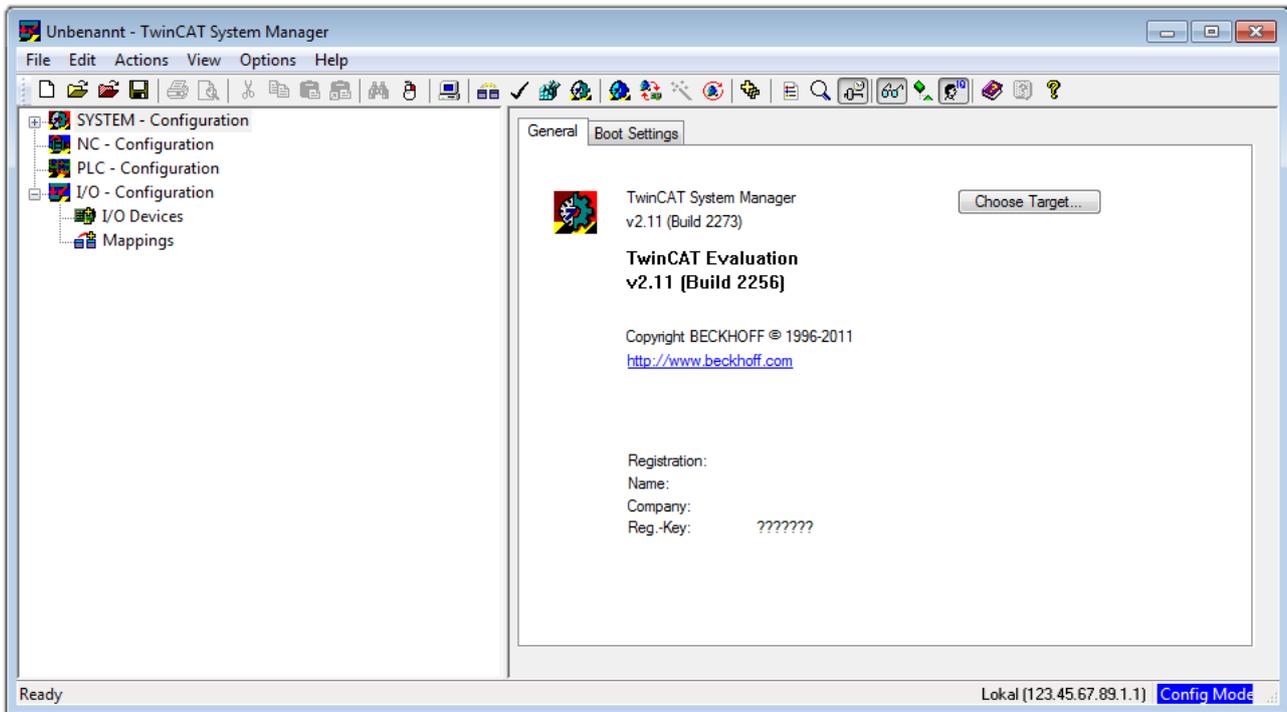


図 26: デフォルトのTwinCAT 2ユーザインターフェイス

通常、TwinCATはローカルまたはリモートモードで使用可能です。ユーザインターフェイス(標準)を含むTwinCATシステムを対象のPLCにインストールすると、TwinCATがローカルモードで使用可能になります。ここで、次のステップの「[デバイスの挿入 \[▶ 39\]](#)」に進みます。

PLC上に構築されたTwinCATランタイム環境を他のシステムからリモートでアドレス指定する場合は、先にタ

ーゲットシステムが認識されている必要があります。アイコン  または「F8」キーを使用して、メニュー[Actions] → [Choose Target System...]と進み、以下のウィンドウを開きます。

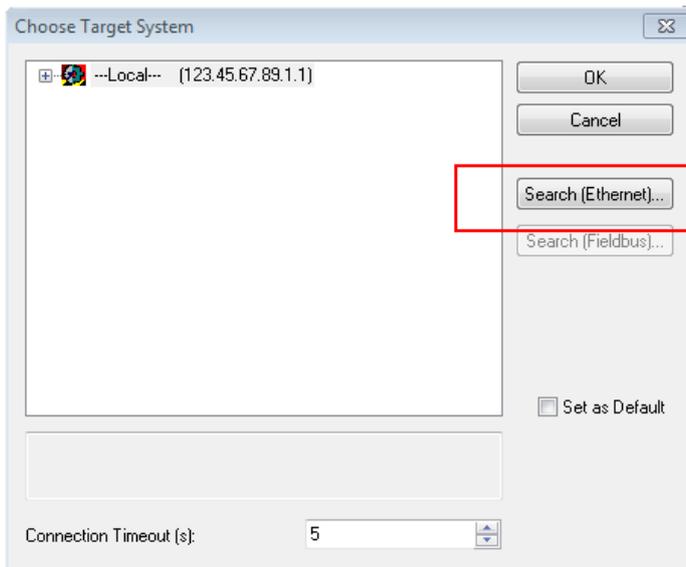


図 27: ターゲットシステムの選択

[Search (Ethernet)...]を使用して、ターゲットシステムを入力します。これにより、以下のいずれかを行うダイアログが開きます。

- ・ [Enter Host Name / IP:] (赤枠)の後に既知のコンピュータ名を入力
- ・ [Broadcast Search]を実行(正確なコンピュータ名が不明な場合)
- ・ 既知のコンピュータのIPアドレスまたはAMSNetIDを入力

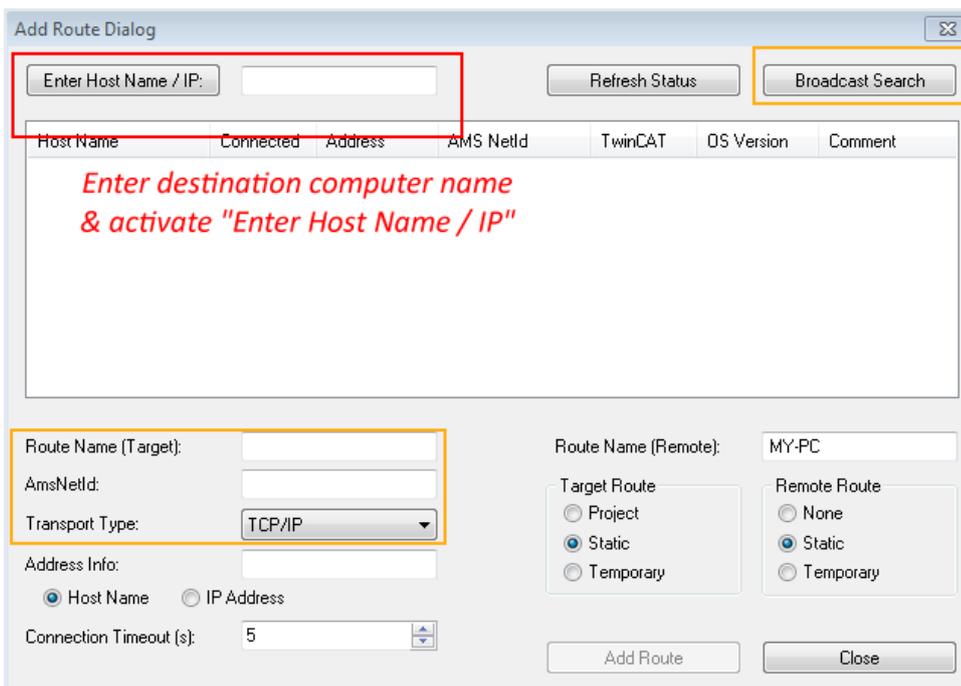
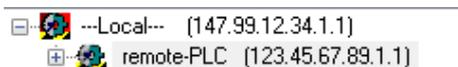


図 28: アクセスするPLCをTwinCAT System Managerを使用して指定: ターゲットシステムの選択

一度ターゲットシステムと接続設定を行うと、次のように選択できます(パスワードの入力が必要な場合があります)。



[OK]で確定後、System Manager経由でターゲットシステムにアクセスできます。

デバイスの追加

TwinCAT 2 System Manager ユーザーインターフェイスの左側にあるコンフィグレーションツリーで [I/O Devices] を選択し、右クリックでコンテキストメニューを開いて [Scan Devices...] を選択するか、メニューバー内の  を使用してスキャンを開始します。TwinCAT System Manager が「Configモード」ではない場合は、 またはメニュー [Actions] → [Set/Reset TwinCAT to Config Mode...] (Shift + F4) を使用して「Configモード」に設定する必要があります。

ユーザー内の  を使用してスキャンを開始します。TwinCAT System Manager が「Configモード」ではな

い場合は、 またはメニュー [Actions] → [Set/Reset TwinCAT to Config Mode...] (Shift + F4) を使用して「Configモード」に設定する必要があります。

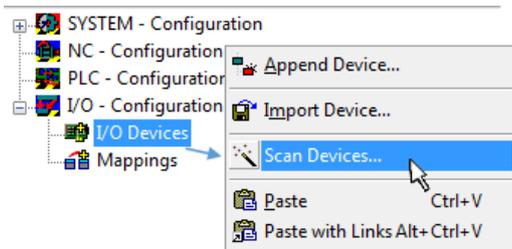


図 29: [Scan Devices...] の選択

表示される警告メッセージを確認し、ダイアログ内で [EtherCAT] を選択します。

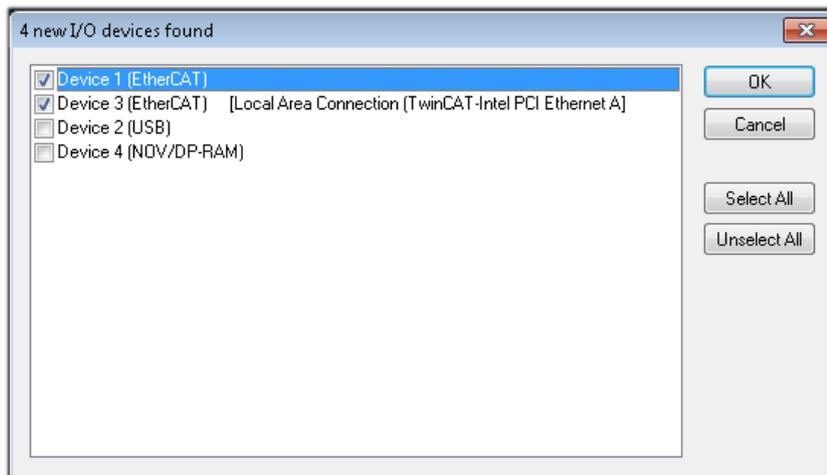


図 30: I/Oデバイスの自動検出: 統合するデバイスの選択

メッセージ「Find new boxes」の確定後、デバイスに接続したターミナルが認識されます。「FreeRun」にすると、「Configモード」での入出力値の操作が可能になります。

本セクションの冒頭に記載した [コンフィグレーション例](#) [▶ 36] の場合、以下の結果となります。

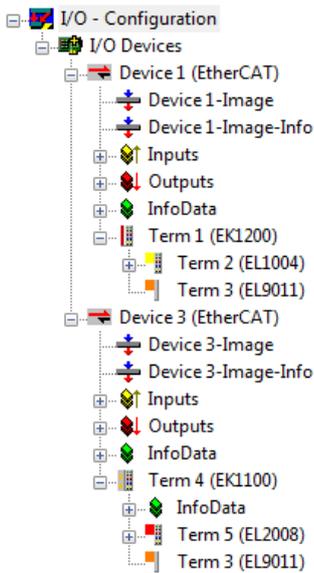


図 31: TwinCAT 2 System Managerでのコンフィギュレーションのマッピング

全プロセスが2段階で構成されます。これらは個別に実行することができます(最初にデバイスを決定し、次にボックス、ターミナルなどの接続するエレメントを決定)。**[Device ...]**コンテキストメニューから選択してスキャンを開始することも可能です。スキャンにより、以下のコンフィギュレーション内に存在するターミナルが読み取られます。

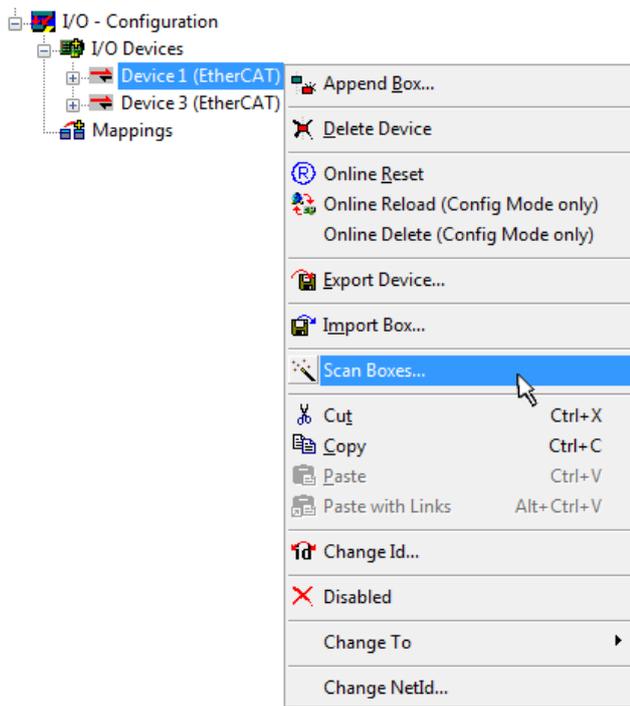


図 32: デバイスに接続されている個々のターミナルのスキャン

この機能は、実際の構成をすぐに変更する場合に便利です。

PLCのプログラミングと組込み

TwinCAT PLC Controlは、複数のプログラム環境内でのコントローラ作成が可能な開発環境です。TwinCAT PLC Controlは、IEC 61131-3に記載されているすべての言語をサポートしています。2つのテキストベースの言語と3つのグラフィカル言語が使用できます。

- ・ テキストベースの言語
 - インストラクションリスト(IL)
 - ストラクチャードテキスト(ST)
- ・ グラフィカル言語

- ファンクションブロックダイアグラム (FBD)
- ラダーダイアグラム (LD)
- コンティニューアスファンクションチャート (CFC)
- シーケンシャルファンクションチャート (SFC)

以下のセクションでは、ストラクチャードテキスト (ST) について説明します。

TwinCAT PLC Controlの開始後、以下のユーザインターフェイスが初期プロジェクトに表示されます。

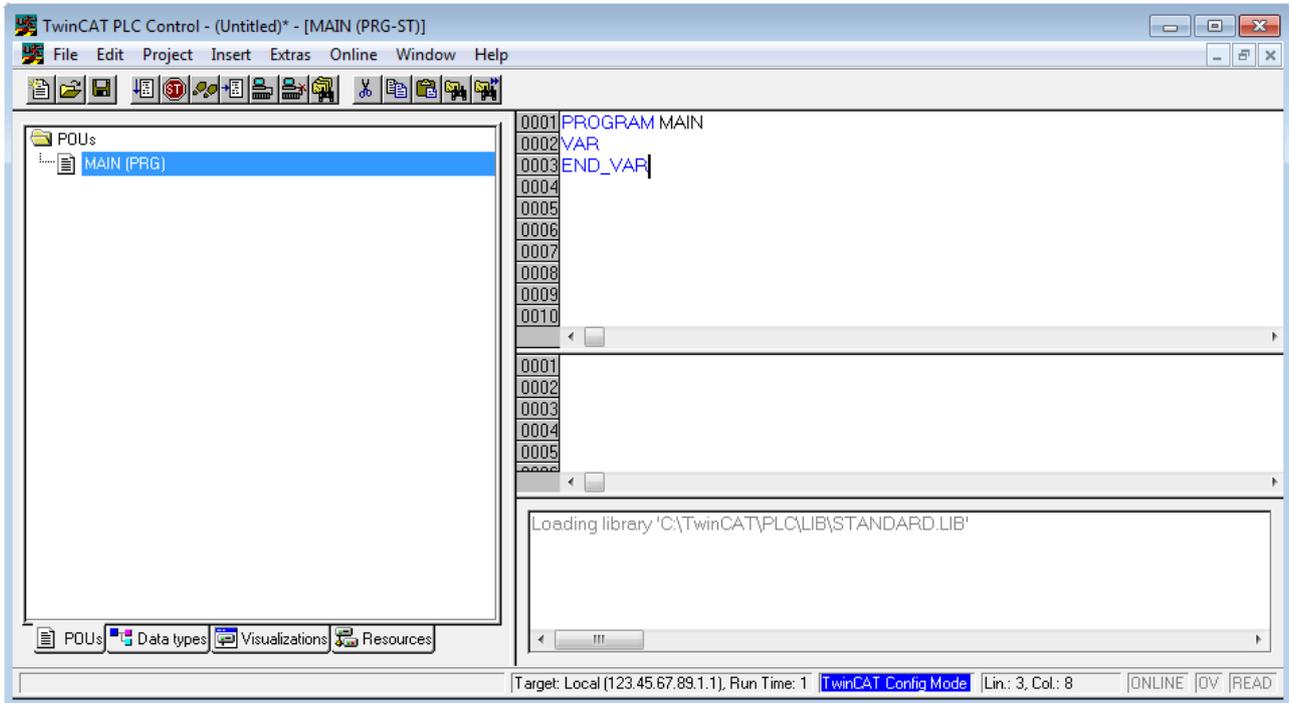


図 33: スタートアップ後のTwinCAT PLC Control

サンプル変数およびサンプルプログラムが作成され、「PLC_example.pro」という名前で保存します。

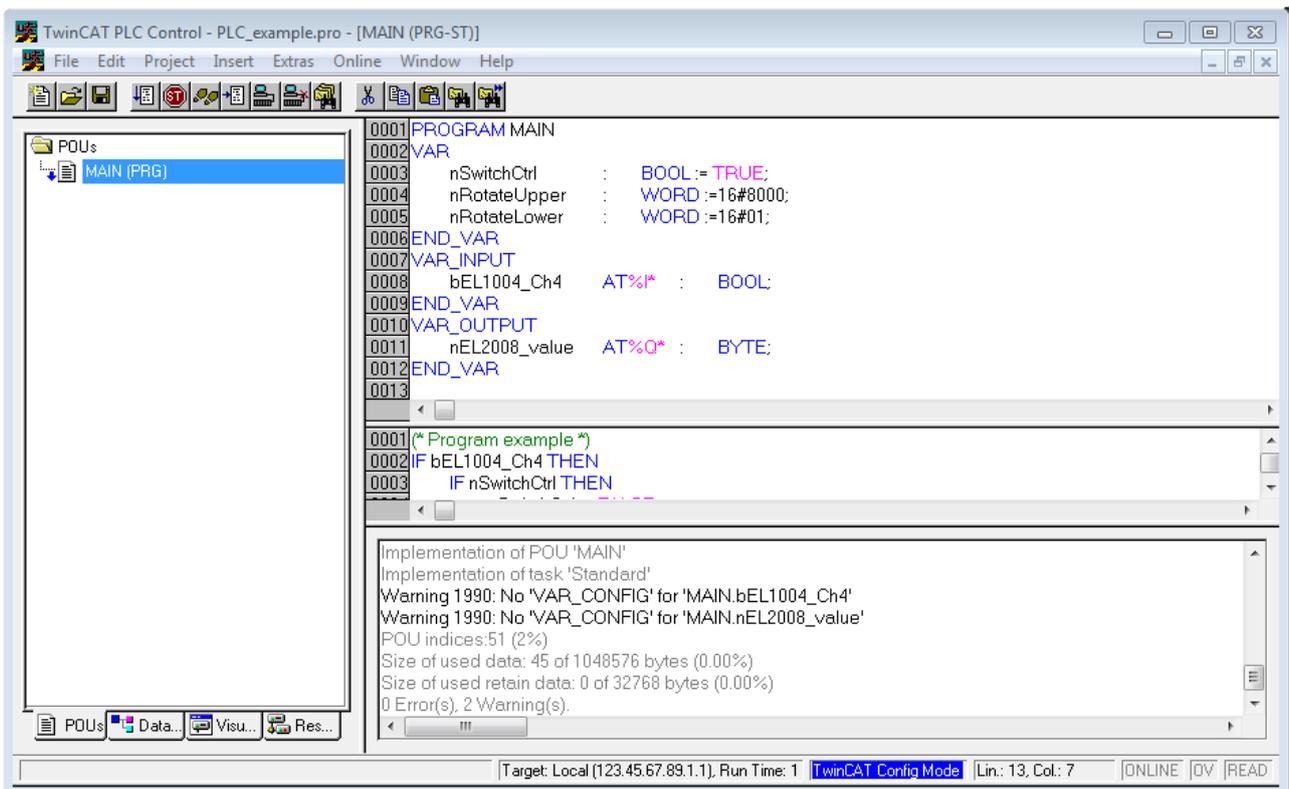


図 34: コンパイル処理後の変数付きのサンプルプログラム(変数統合なし)

コンパイル処理後の警告1990（「VAR_CONFIG」の欠落）は、外部（ID「AT%I*」または「AT%Q*」）として定義した変数が割り当てられていないことを示しています。正常にコンパイルされると、TwinCAT PLC Controlはプロジェクトが保存されたディレクトリに「*.tpy」ファイルを作成します。このファイル（*.tpy）には変数割り当てが含まれており、System Managerには認識されていないため、警告が表示されます。System Managerに通知されると、警告は表示されなくなります。

最初に、PLC - Configurationのコンテキストメニューを使用して、TwinCAT PLC ControlプロジェクトをSystem Managerに追加します。右クリックして[Append PLC Project...]を選択します。

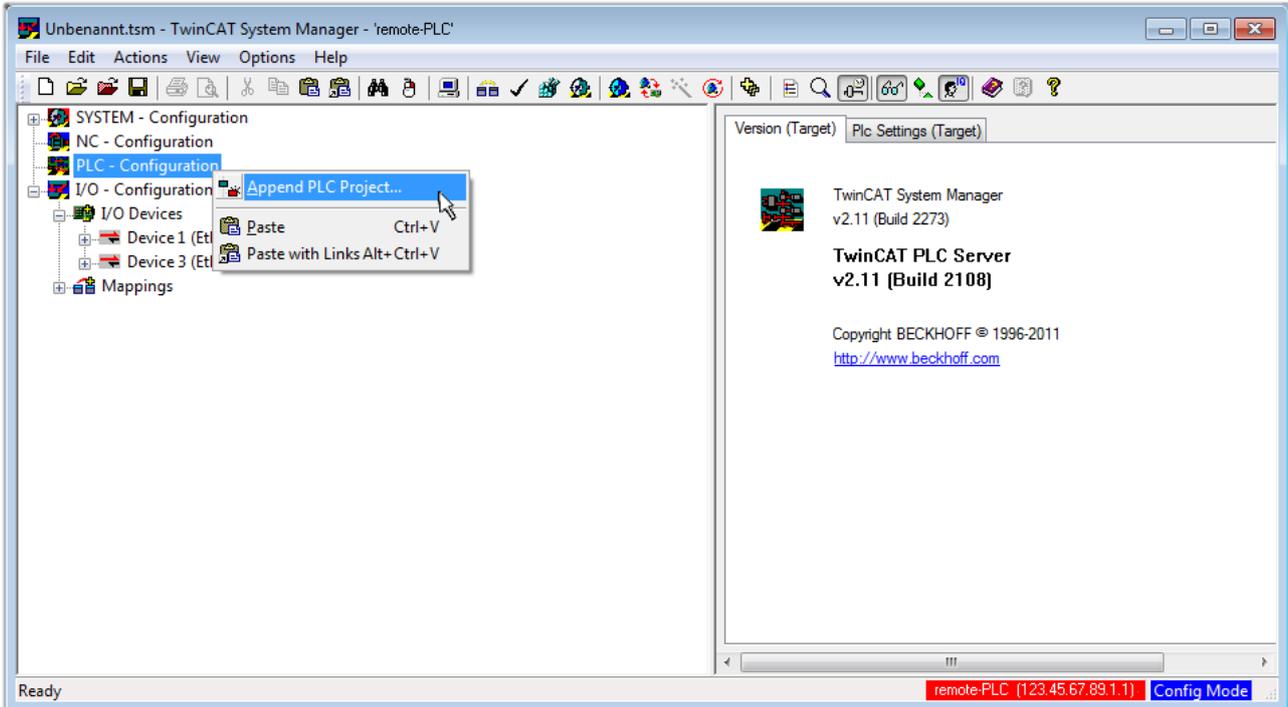


図 35: TwinCAT PLC Controlプロジェクトの追加

開いたブラウザウィンドウ内で、PLCコンフィグレーション「PLC_example.tpy」を選択します。「AT」で識別される2つの変数を含むプロジェクトが、System Managerのコンフィグレーションツリーに追加されます。

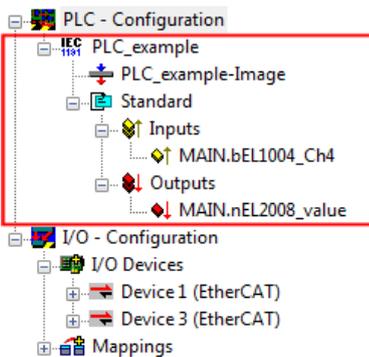


図 36: System ManagerのPLCコンフィグレーションに追加されたPLCプロジェクト

これで、2つの変数「bEL1004_Ch4」と「nEL2008_value」が、I/O設定の特定のプロセスオブジェクトに割り当てられます。

変数の割り当て

追加したプロジェクト「PLC_example」の変数のコンテキストメニュー、および[Modify Link...] → [Standard]から、適切なプロセスオブジェクト(PDO)を選択するためのウィンドウを開きます。

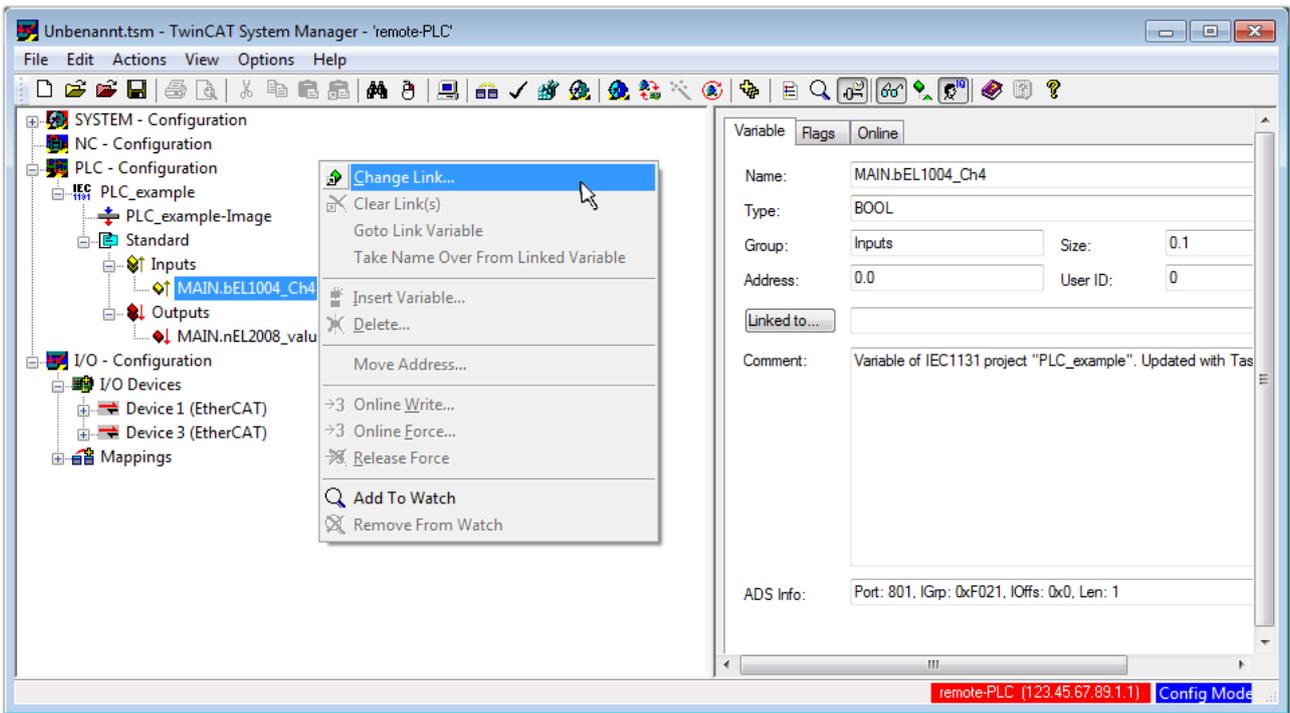


図 37: PLC変数とプロセスオブジェクト間のリンクの作成

開いたウィンドウ内で、PLC - ConfigurationツリーからBOOL型の変数「bEL1004_Ch4」のプロセスオブジェクトを選択できます。

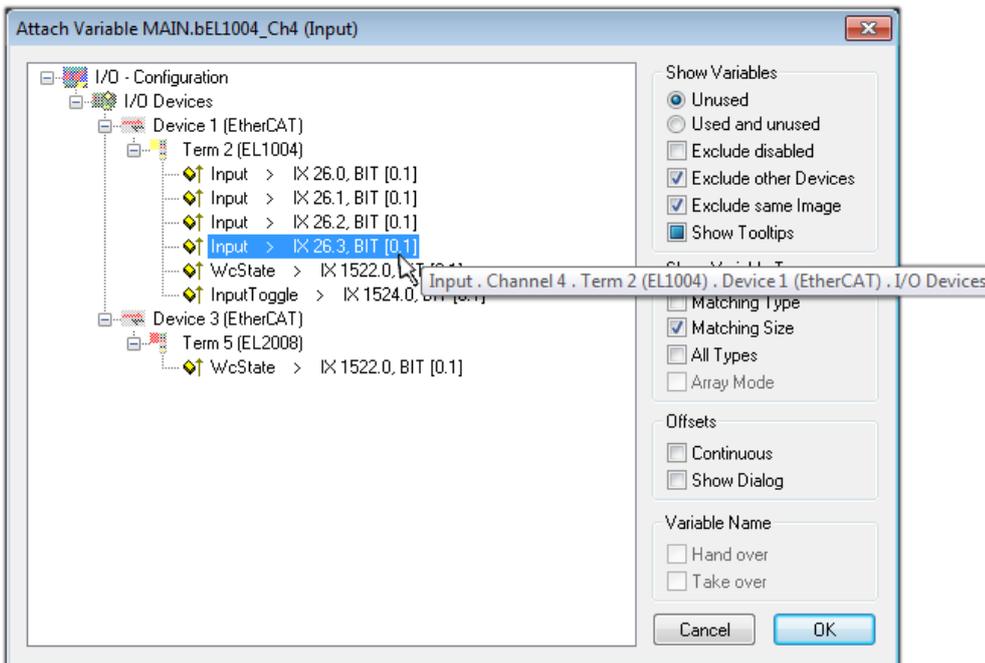


図 38: BOOL型のPDOの選択

デフォルト設定の場合は、選択可能なPDOオブジェクトがここで表示されます。このサンプルでは、EL1004ターミナルのチャンネル4の入力をリンク用を選択しています。もう一方は8つの個々の出力ビットをバイト変数に割り当てるため、出力変数のリンク作成用にチェックボックス[All types]をチェックする必要があります。以下の図は、プロセス全体を表しています。

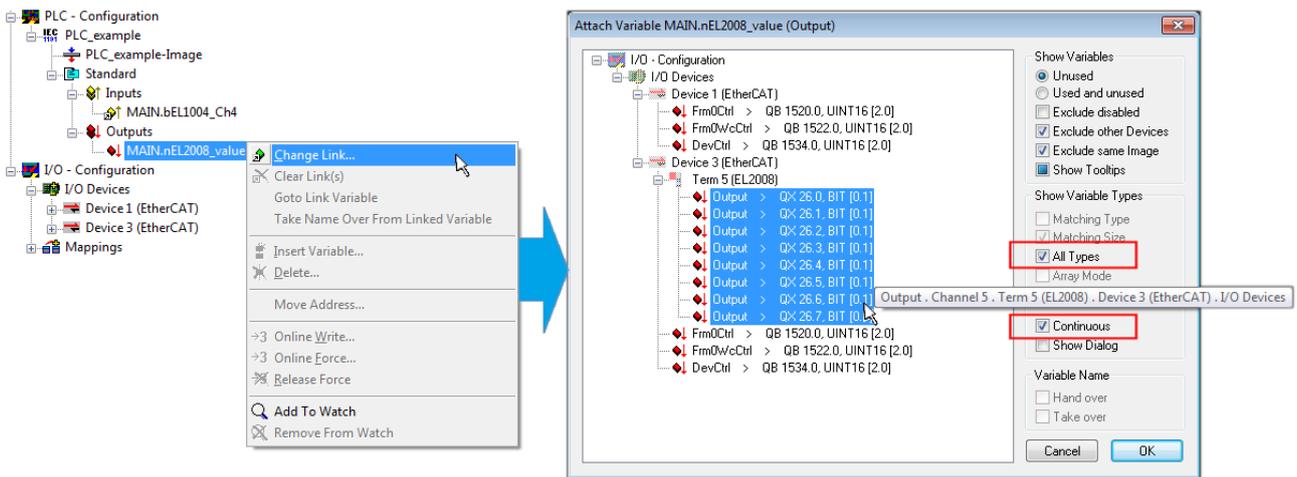


図 39: 複数のPDOの同時選択: [Continuous]および[All types]を有効化

[Continuous]チェックボックスも有効にすることに注意してください。このチェックボックスは、変数「nEL2008_value」のバイト内に含まれるビットをEL2008ターミナルの選択された8つの出力ビットすべてに順番に割り当てるために有効にします。これにより、PLCのチャンネル1に対応するビット0からチャンネル8に対応するビット7までをバイト型で、プログラム内でターミナルの8つの出力すべてを順番にアドレス指定できます。変数の黄または赤のオブジェクトに表示されている特殊なアイコン(Ⓜ)は、リンクが存在していることを示しています。リンクは、変数のコンテキストメニューから[Goto Link Variable]を選択してチェックすることもできます。対応するオブジェクト(ここではPDO)が、自動的に選択されます。

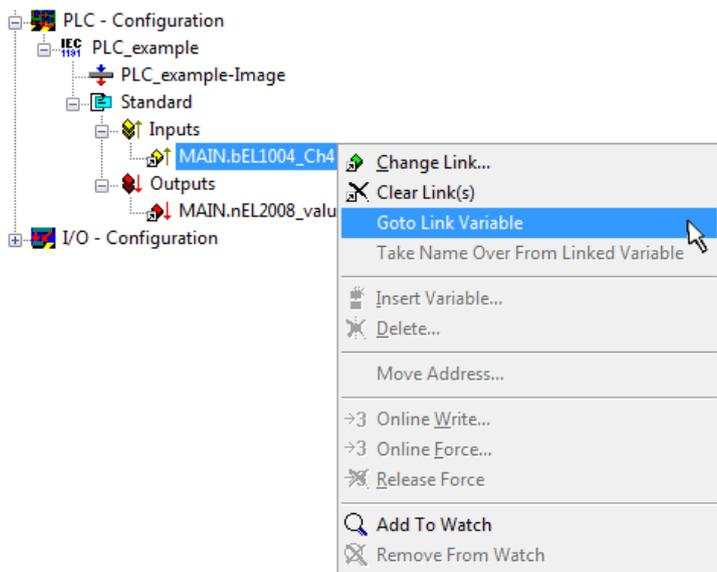


図 40: 「MAIN.bEL1004_Ch4」をサンプルとして使用した、「Goto Link」変数の使用例

変数をPDOに割り当てる処理は、メニュー[Actions] → [Generate Mappings]の選択、Ctrl+Mキー、または

メニュー内のアイコン  を使用して行います。

PDOの割り当ては、コンフィグレーション内で可視化することが可能です。



リンク作成の処理は、上記と逆方向(個々のPDOから変数へ)でも行えます。ただし、この例ではターミナルがデジタル出力を個々にしか有効にしないため、EL2008に対する出力ビットをすべて選択することはできません。ターミナルにバイト、ワード、整数、または同様のPDOがある場合は、これをビット標準化した変数(「BOOL」型)のセットに割り当てられます。この場合も、PDOのコンテキストメニューの[Goto Link Variable]で逆方向に実行し、各PLCインスタンスを選択することが可能です。

コンフィグレーションの有効化

PDOをPLC変数への割り当てることで、コントローラからターミナルの入出力への接続が確立されます。これ

により、この設定を有効にすることが可能になります。最初に、設定を  (または[Actions] →

[Check Configuration])で検証できます。エラーがない場合は、設定を  (または[Actions] → [Activate Configuration...])で有効にし、System Managerでの設定をランタイムシステムに転送できます。「Old configurations are overwritten!」および「Restart TwinCAT system in Run mode」というメッセージを確認し、[OK]で確定します。

数秒後、リアルタイムステータス **RTime 0%** がSystem Managerの右下に表示されます。これにより、PLCシステムを以下の説明にしたがって開始できます。

コントローラの開始

リモートシステムから開始する場合は、[Online] → [Choose Run-Time System...]からPLC制御をイーサネット経由で組込み型PCとリンクさせる必要があります。

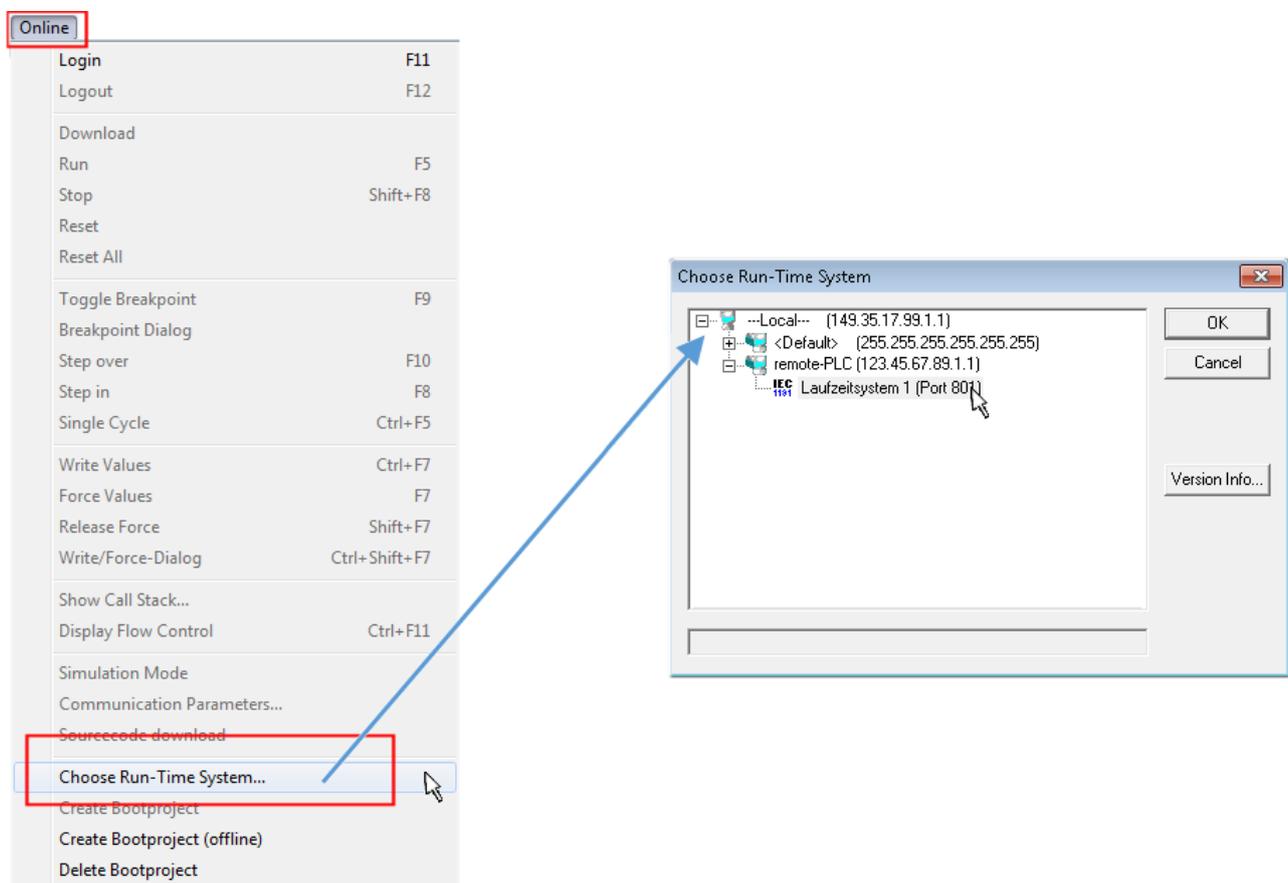


図 41: ターゲットシステムの選択(リモート)

このサンプルでは、「Runtime system 1 (port 801)」を選択し、確定しています。メニューオプション

[Online] → [Login]、F11キー、またはアイコン  のクリックで、制御プログラムをリアルタイムシステムとリンクします。リンク後、制御プログラムを実行用にロードできます。表示されるメッセージ「No program on the controller! Should the new program be loaded?」に対して、[Yes]で確定します。これで、ランタイム環境でプログラムを開始することができます。

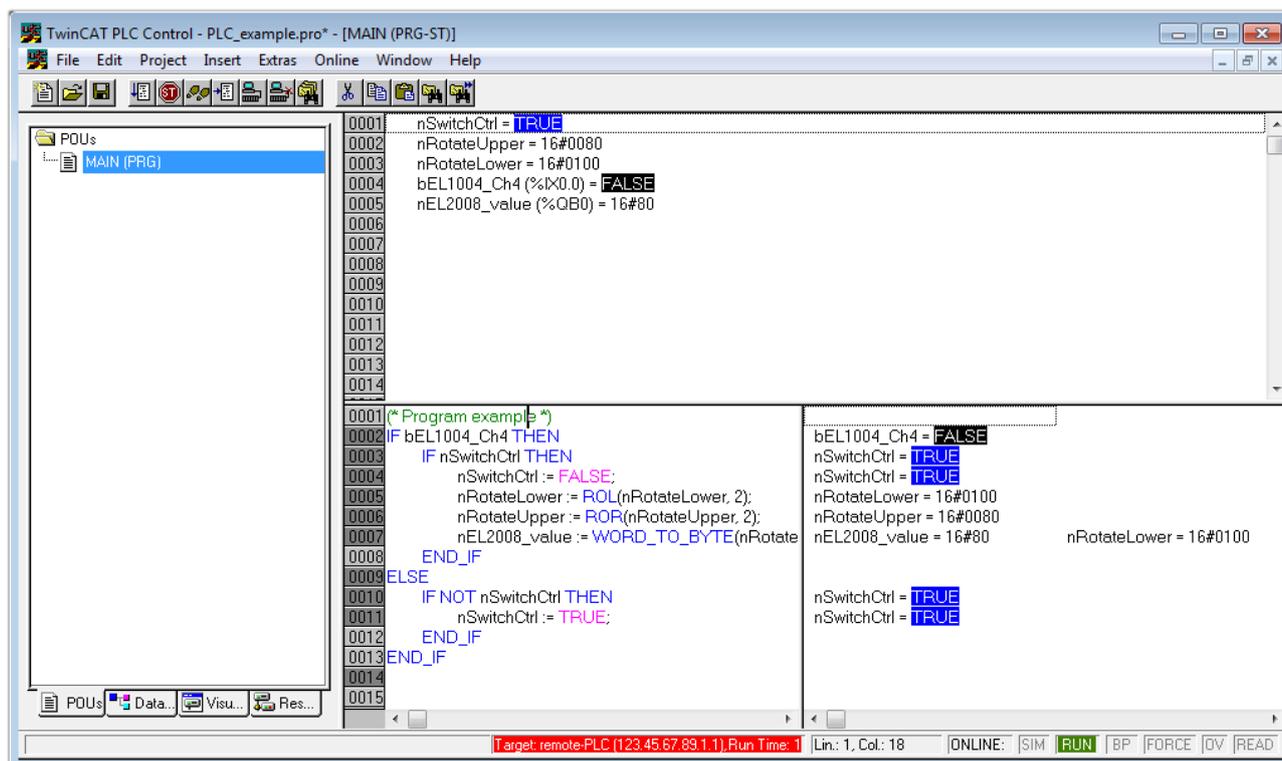


図 42: PLC制御にログイン、プログラムを開始可能

[Online] → [Run]、F5キー、または  でPLCを開始できます。

5.1.2 TwinCAT 3

スタートアップ

TwinCATには、Microsoft Visual Studioを使用した開発環境エリアが用意されています。スタートアップ後、一般ウィンドウエリア(参考: TwinCAT 2の「TwinCAT System Manager」)の左側に、電気機械的コンポーネントとの通信向けのプロジェクトフォルダエクスプローラが表示されます。

開発に使用するPCにTwinCATシステムを正常にインストールすると、スタートアップ後にTwinCAT 3 (シェル)が以下のユーザインターフェイスを表示します。



図 43: デフォルトのTwinCAT 3ユーザインターフェイス

最初に、 **New TwinCAT Project...** (または[File] → [New] → [Project...])を使用してプロジェクトを新規作成します。以下のダイアログで、必要に応じて入力を行います(図を参照)。

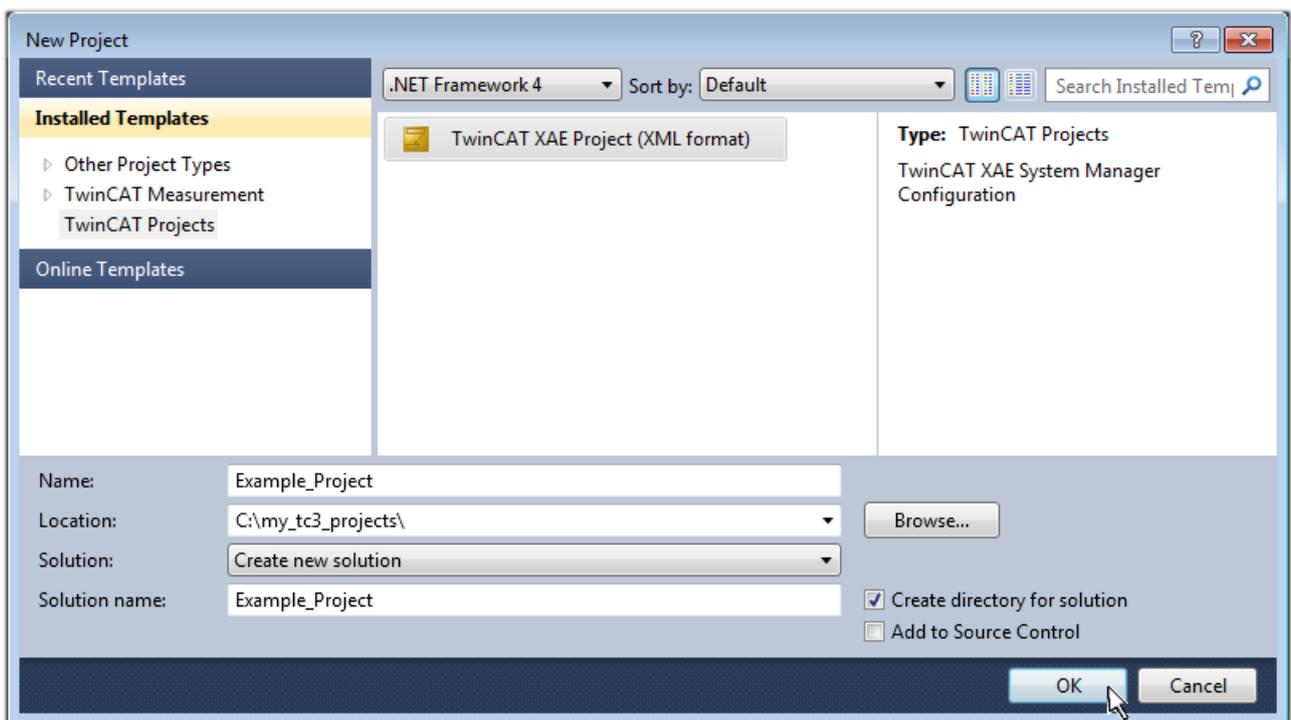


図 44: TwinCATプロジェクトの新規作成

これで、プロジェクトフォルダエクスプローラ内に新規プロジェクトが作成されます。

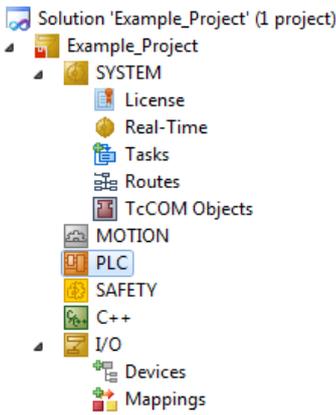
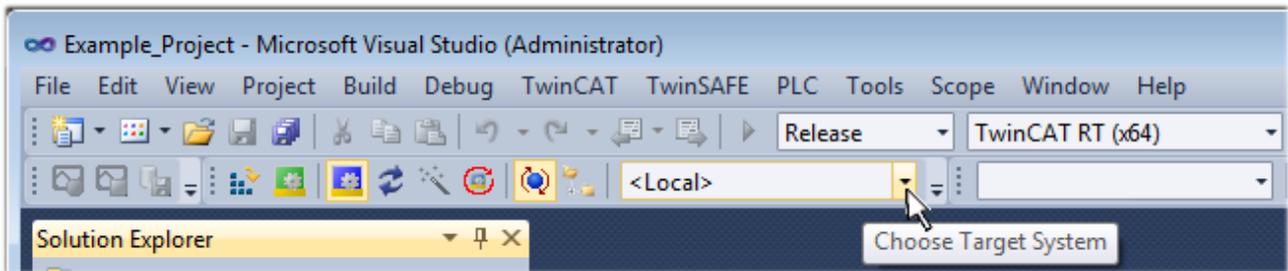


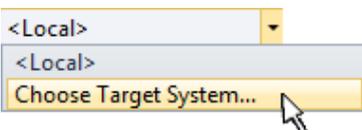
図 45: プロジェクトフォルダエクスプローラ内の新規TwinCAT3プロジェクト

通常、TwinCATはローカルまたはリモートモードで使用可能です。ユーザインターフェイス(標準)を含むTwinCATシステムが対象のPLCにインストールされると、TwinCATがローカルモードで使用可能になります。ここで、次のステップの「デバイスの挿入」[▶ 49]に進みます。

PLC上に構築したTwinCATランタイム環境を他のシステムからリモートでアドレス指定する場合は、先にターゲットシステムが認識されている必要があります。メニューバー内のアイコンを使用し、



プルダウンメニューを展開して



以下のウィンドウを開きます。

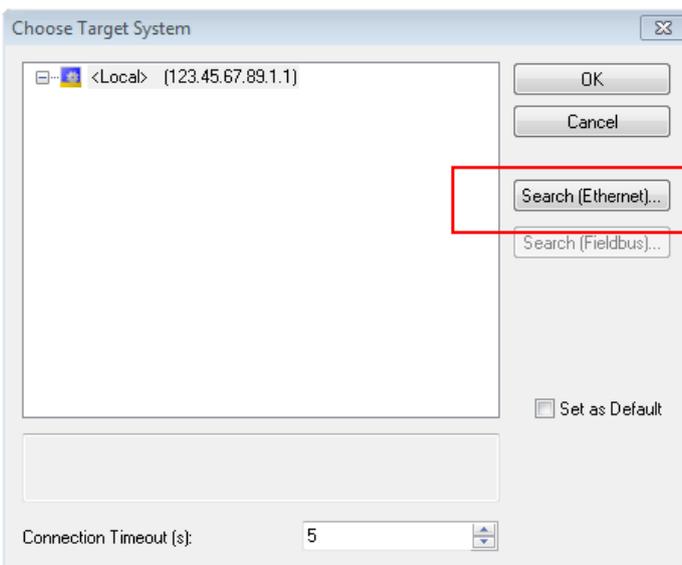


図 46: 選択ダイアログ: ターゲットシステムの選択

[Search (Ethernet)...]を使用して、ターゲットシステムを入力します。これにより、以下のいずれかを行うダイアログが開きます。

- ・ [Enter Host Name / IP:] (赤枠)の後に既知のコンピュータ名を入力
- ・ [Broadcast Search]を実行(正確なコンピュータ名が不明な場合)
- ・ 既知のコンピュータのIPアドレスまたはAMSNetIDを入力

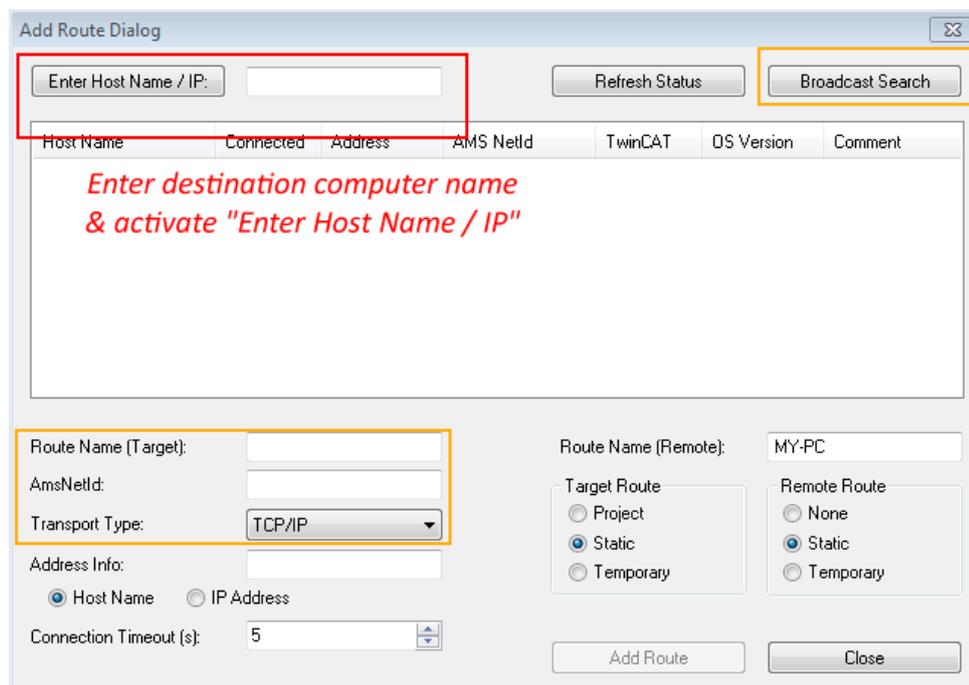
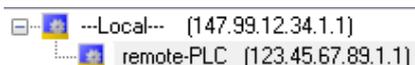


図 47: アクセスするPLCをTwinCAT System Managerを使用して指定: ターゲットシステムの選択

一度ターゲットシステムと接続設定を行うと、次のように選択できます(パスワードの入力が必要な場合があります)。



[OK]で確定後、Visual Studioシェル経由でターゲットシステムにアクセスできます。

デバイスの追加

Visual Studioシェルユーザインターフェイスのプロジェクトフォルダエクスプローラで、エレメント[I/O]内の[Devices]を選択し、右クリックでコンテキストメニューを開いて[Scan]を選択するか、メニューバー

内の  を使用してスキャンを開始します。TwinCAT System Managerが「Configモード」ではない場合

は、 またはメニュー[TwinCAT] → [Restart TwinCAT (Config mode)] を使用して「Configモード」に設定する必要があります。

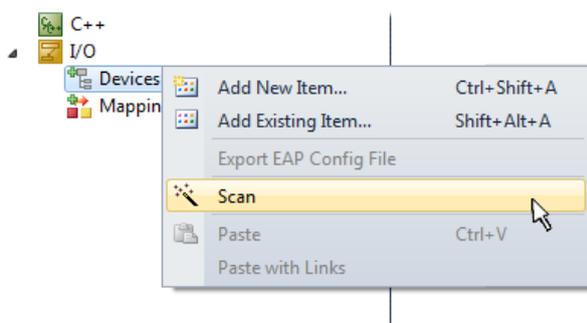


図 48: [Scan]の選択

表示される警告メッセージを確認し、ダイアログ内で[EtherCAT]を選択します。

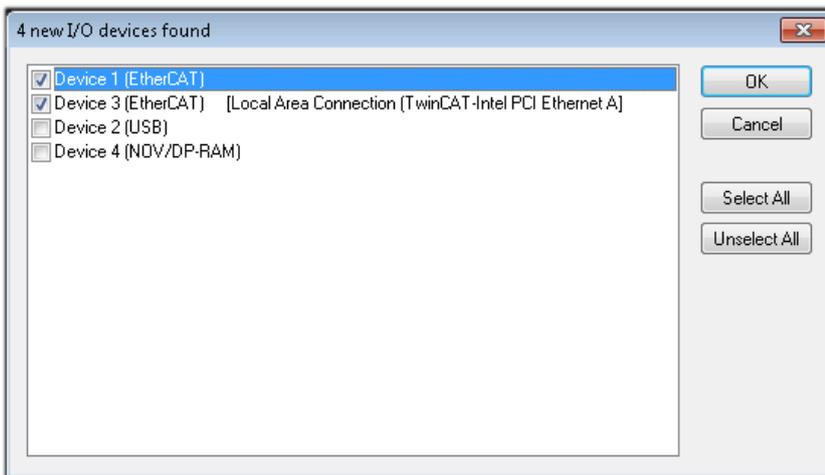


図 49: I/Oデバイスの自動検出: 統合するデバイスの選択

メッセージ「Find new boxes」の確定後、デバイスに接続したターミナルが認識されます。「FreeRun」にすると、「Configモード」での入出力値の操作が可能になります。

本セクションの冒頭に記載したコンフィグレーション例 [▶ 36]の場合、以下の結果となります。

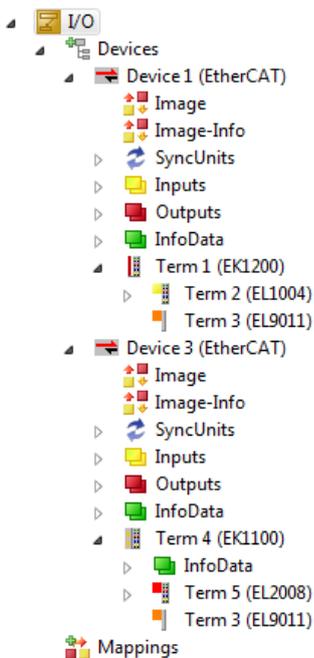


図 50: TwinCAT 3環境のVSシェルでのコンフィグレーションのマッピング

全プロセスが2段階で構成されます。これらは個別に実行することができます(最初にデバイスを決定し、次にボックス、ターミナルなどの接続するエレメントを決定)。「[Device ...]」コンテキストメニューから選択してスキャンを開始することも可能です。スキャンにより、以下のコンフィグレーション内に存在するターミナルが読み取られます。

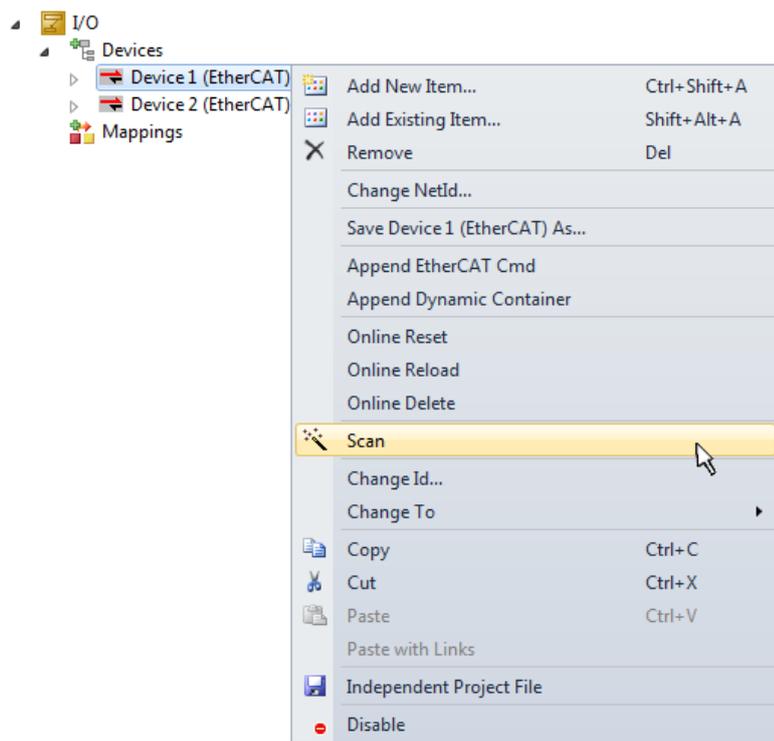


図 51: デバイスに接続されている個々のターミナルのスキャン

この機能は、実際の構成をすぐに変更する場合に便利です。

PLCのプログラミング

TwinCAT PLC Controlは、複数のプログラム環境内でのコントローラ作成が可能な開発環境です。TwinCAT PLC Controlは、IEC 61131-3に記載されているすべての言語をサポートしています。2つのテキストベースの言語と3つのグラフィカル言語が使用できます。

- ・ **テキストベースの言語**
 - インストラクションリスト (IL)
 - ストラクチャードテキスト (ST)
- ・ **グラフィカル言語**
 - ファンクションブロックダイアグラム (FBD)
 - ラダーダイアグラム (LD)
 - コンティニューアスファンクションチャート (CFC)
 - シーケンシャルファンクションチャート (SFC)

以下のセクションでは、ストラクチャードテキスト (ST) について説明します。

プログラミング環境を作成するために、プロジェクトフォルダエクスプローラ内の [PLC] のコンテキストメニューで [Add New Item...] を選択し、プロジェクトサンプルに PLC サブプロジェクトを追加します。

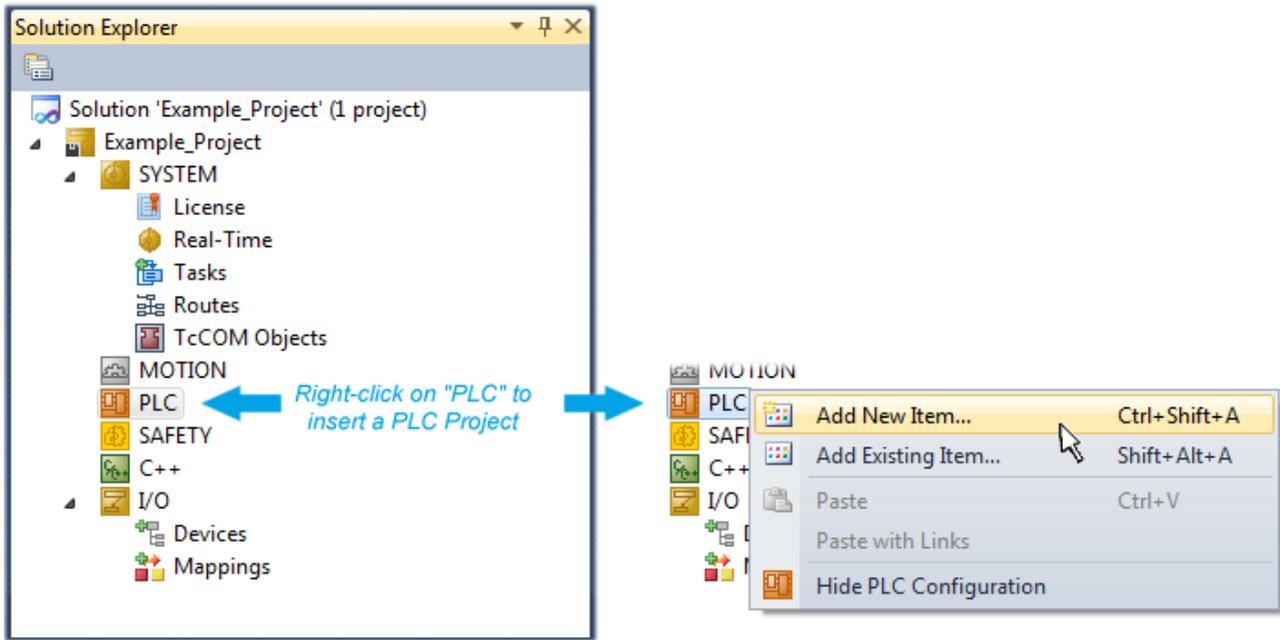


図 52: [PLC]内でのプログラミング環境の追加

開いたダイアログで[Standard PLC project]を選択し、プロジェクト名として例えば「PLC_example」と入力して対応するディレクトリを選択します。

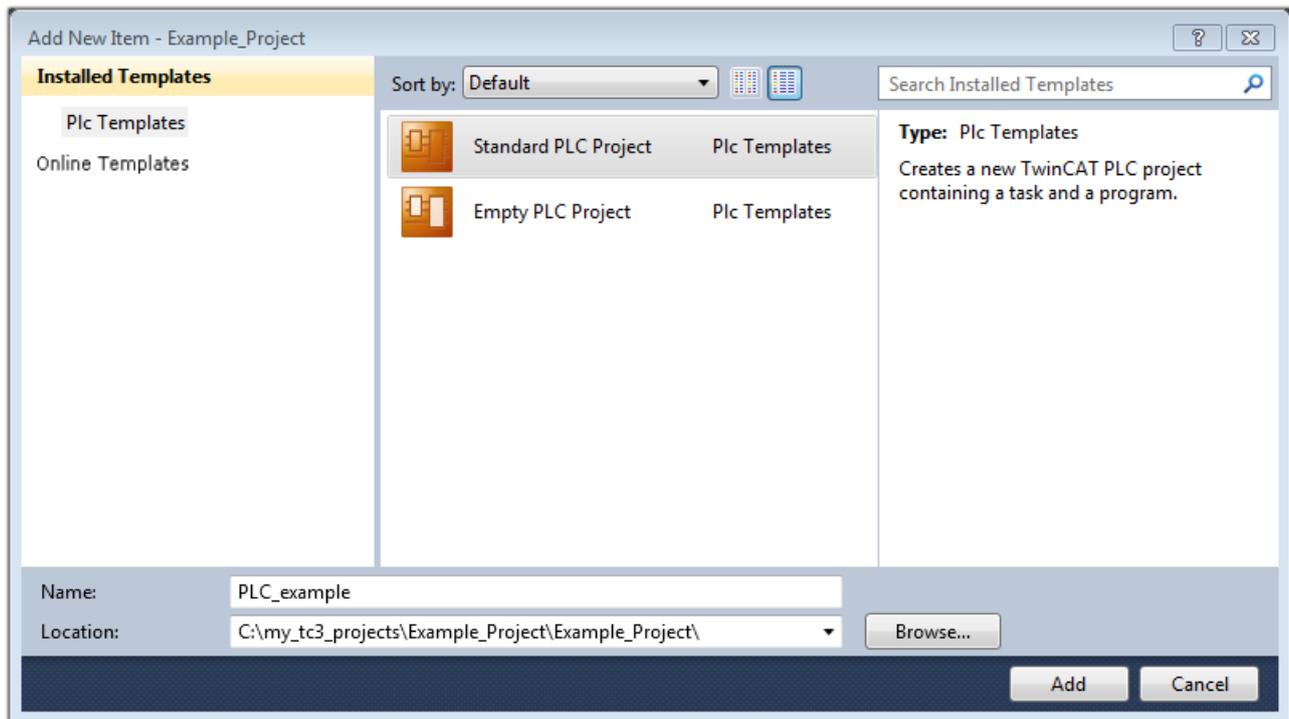


図 53: PLCプログラミング環境の名前およびディレクトリの指定

[Standard PLC project]を選択して作成した「Main」プログラムは、[POUs]内の[PLC_example_project]をダブルクリックすると開けます。初期プロジェクトには、以下のユーザーインターフェイスが表示されます。

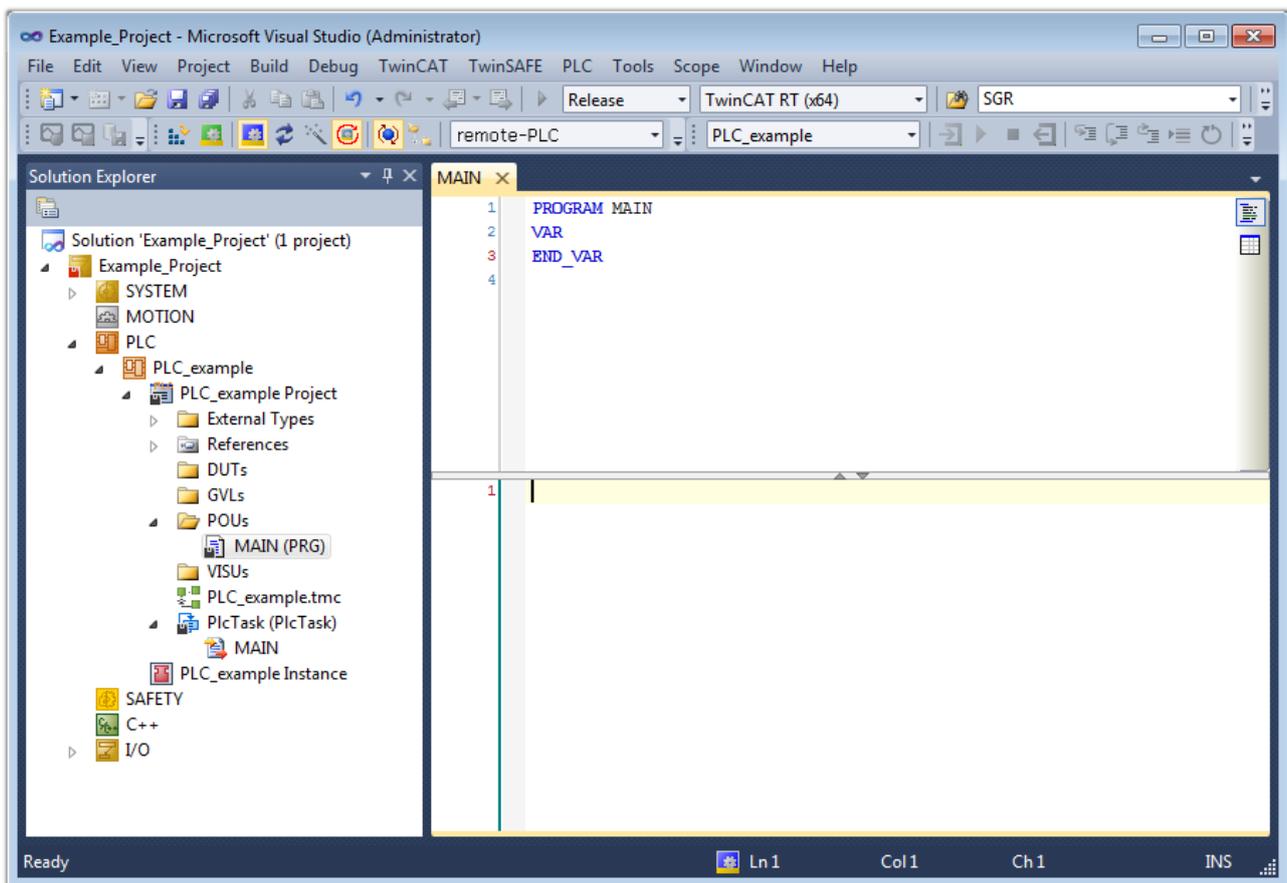


図 54: 標準PLCプロジェクトの初期「Main」プログラム

説明のために、サンプル変数およびサンプルプログラムが作成されています。

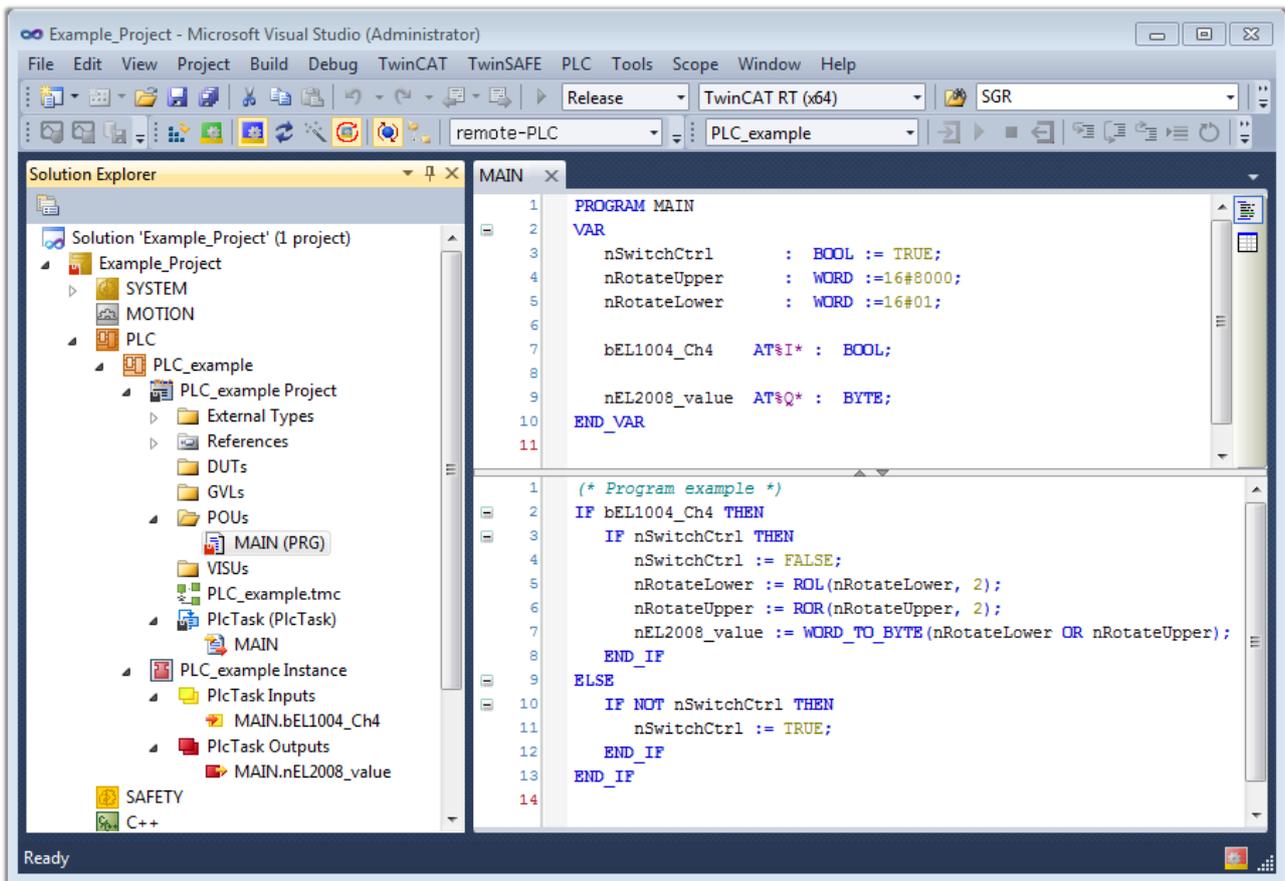


図 55: コンパイル処理後の変数付きのサンプルプログラム(変数統合なし)

コンパイル処理後、制御プログラムがプロジェクトフォルダとして作成されます。

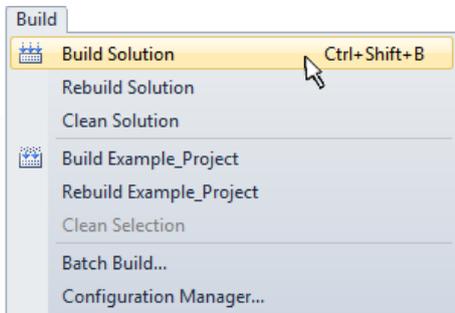
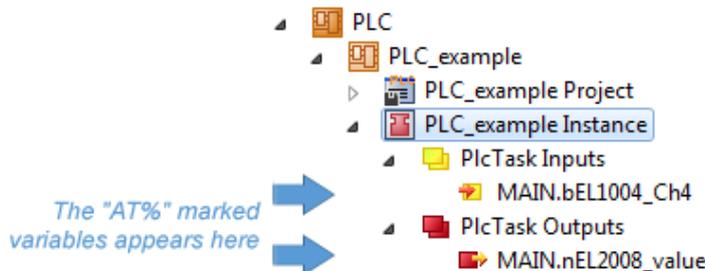


図 56: プログラムのコンパイルの開始

ST/PLCプログラム内において「AT%」で識別される以下の変数が、プロジェクトフォルダエクスプローラの [Assignments] で使用できます。



変数の割り当て

[PLC]のインスタンス→変数のコンテキストメニューから、[Modify Link...]オプションを使用して、リンクする適切なプロセスオブジェクト(PDO)を選択するためのウィンドウを開きます。

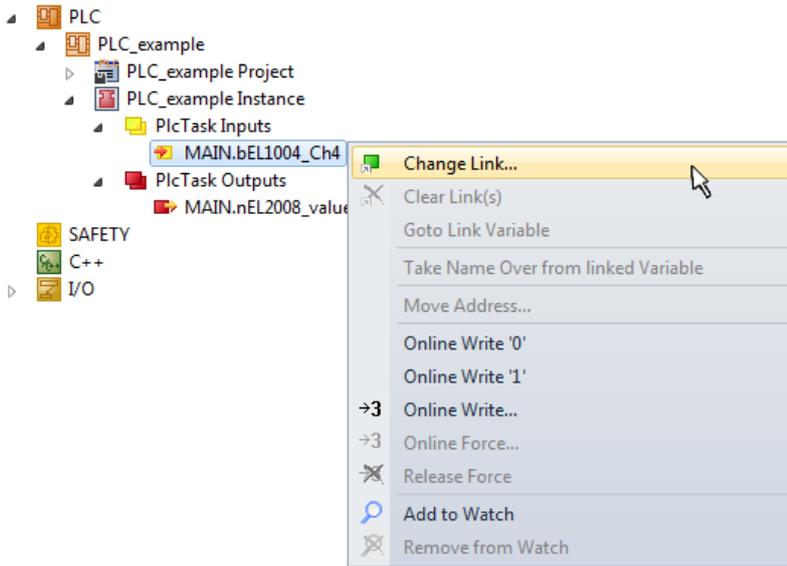


図 57: PLC変数とプロセスオブジェクト間のリンクの作成

開いたウィンドウ内で、PLC - ConfigurationツリーからBOOL型の変数「bEL1004_Ch4」のプロセスオブジェクトを選択できます。

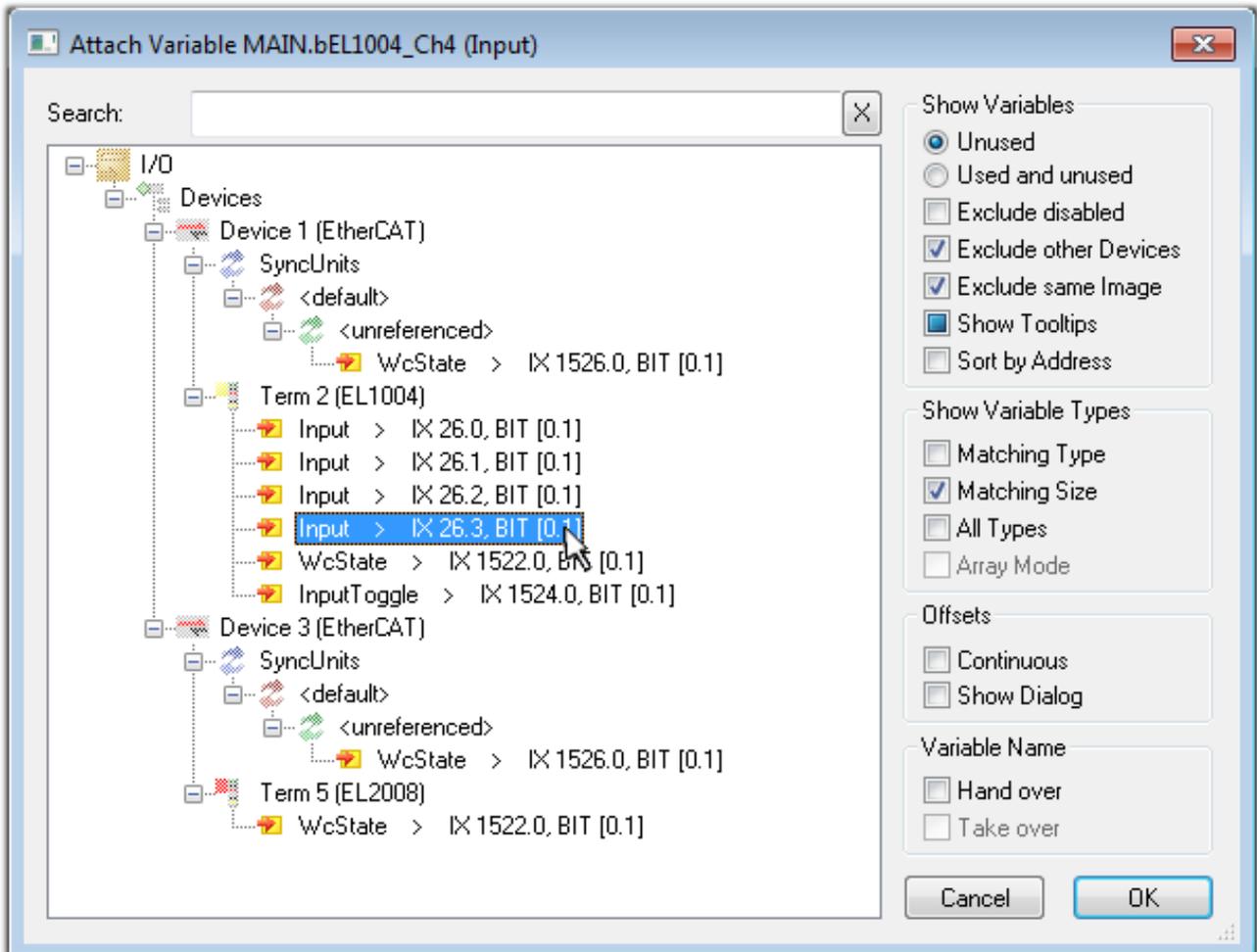


図 58: BOOL型のPDOの選択

デフォルト設定の場合は、選択可能なPDOオブジェクトがここで表示されます。このサンプルでは、EL1004ターミナルのチャンネル4の入力をリンク用に選択しています。もう一方は8つの個々の出力ビットをバイト変数に割り当てるため、出力変数のリンク作成用にチェックボックス[All types]をチェックする必要があります。以下の図は、プロセス全体を表しています。

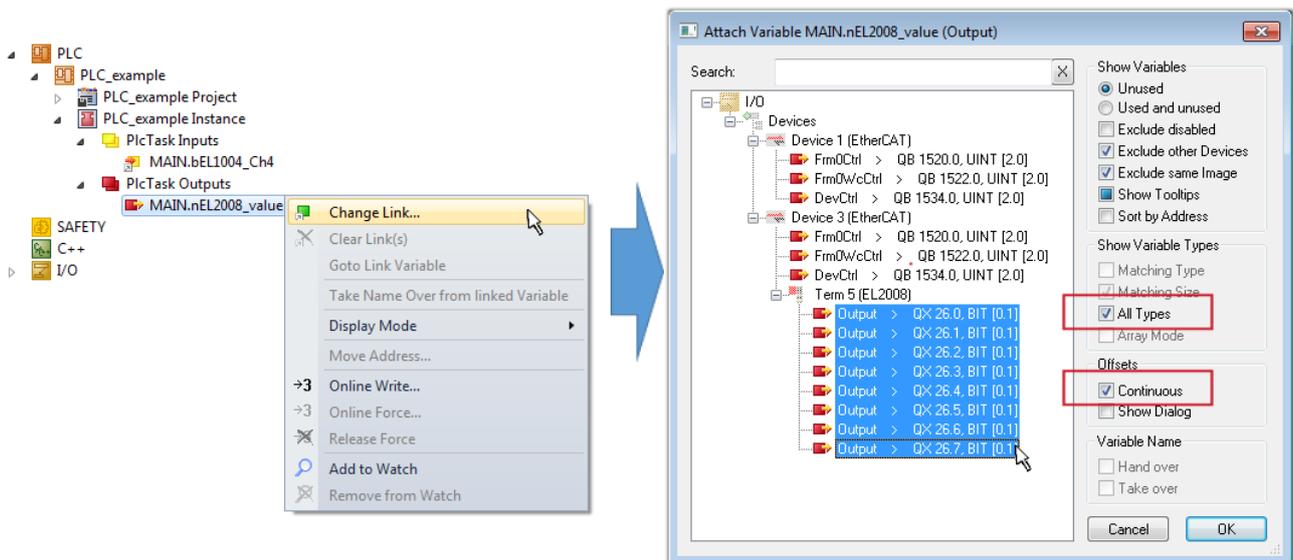


図 59: 複数のPDOの同時選択: [Continuous]および[All types]を有効化

[Continuous]チェックボックスも有効にすることに注意してください。このチェックボックスは、変数「nEL2008_value」のバイト内に含まれるビットをEL2008ターミナルの選択された8つの出力ビットすべてに順番に割り当てるために有効にします。これにより、PLCのチャンネル1に対応するビット0からチャンネル8に対応するビット7までをバイト型で、プログラム内でターミナルの8つの出力すべてを順番にアドレス指定できます。変数の黄または赤のオブジェクトに表示されている特殊なアイコン(🔗)は、リンクが存在していることを示しています。リンクは、変数のコンテキストメニューから[Goto Link Variable]を選択してチェックすることもできます。対応するオブジェクト(ここではPDO)が、自動的に選択されます。

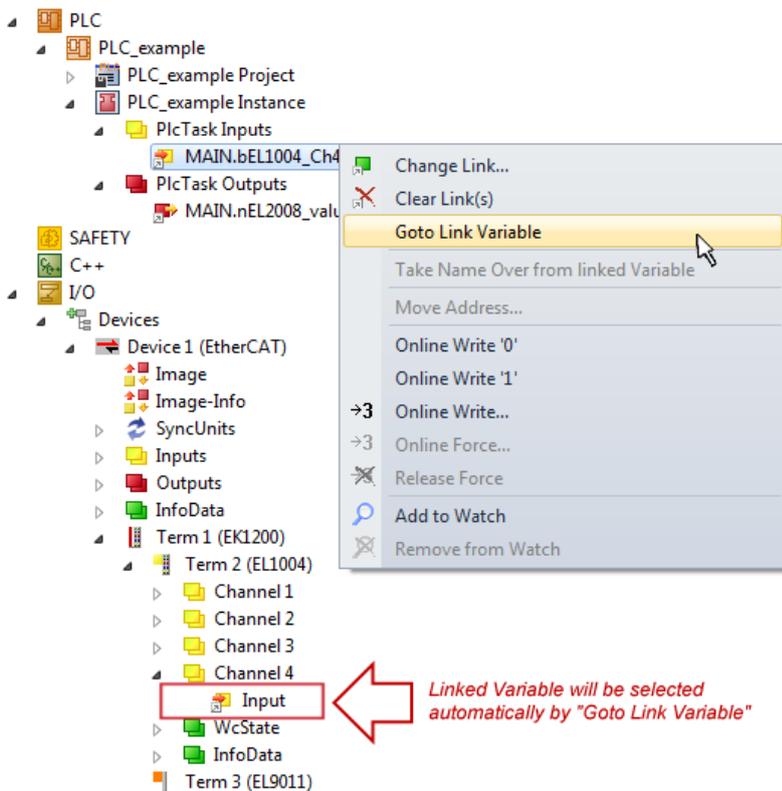


図 60: 「MAIN. bEL1004_ Ch4」をサンプルとして使用した、「Goto Link」変数の使用例

リンク作成の処理は、上記と逆方向(個々のPDOから変数へ)でも行えます。ただし、この例ではターミナルがデジタル出力を個々にしか有効にしないため、EL2008に対する出力ビットをすべて選択することはできません。ターミナルにバイト、ワード、整数、または同様のPDOがある場合は、これをビット標準化した変数(「BOOL」型)のセットに割り当てられます。この場合も、PDOのコンテキストメニューの[Goto Link Variable]で逆方向に実行し、各PLCインスタンスを選択することが可能です。

コンフィグレーションの有効化

PDOのPLC変数への割り当てすることで、コントローラからターミナルの入出力への接続が確立されます。これ

により、 または[TwinCAT]のメニューからコンフィグレーションを有効にして、開発環境の設定をランタイムシステムに転送できます。「Old configurations are overwritten!」および「Restart TwinCAT system in Run mode」というメッセージを確認し、[OK]で確定します。プロジェクトフォルダエクスプローラ内に対応する割り当てが表示されます。

-  Mappings
 -  PLC_example Instance - Device 3 (EtherCAT) 1
 -  PLC_example Instance - Device 1 (EtherCAT) 1

数秒後、Runモードに対応するステータスが、VSシェル開発環境の右下に回転するアイコン  として表示されます。これにより、PLCシステムを以下の説明にしたがって開始できます。

コントローラの開始

メニューオプション[PLC] → [Login]の選択、または  のクリックで、PLCをリアルタイムシステムとリンクし、実行する制御プログラムをロードします。表示されるメッセージ「No program on the

controller! Should the new program be loaded?」に対して、[Yes]で確定します。アイコン  のクリック、「F5」キー、または[PLC]メニューの[Start]で、ランタイム環境でプログラムを開始することができます。開始されたプログラミング環境には、各変数のランタイム値が表示されます。

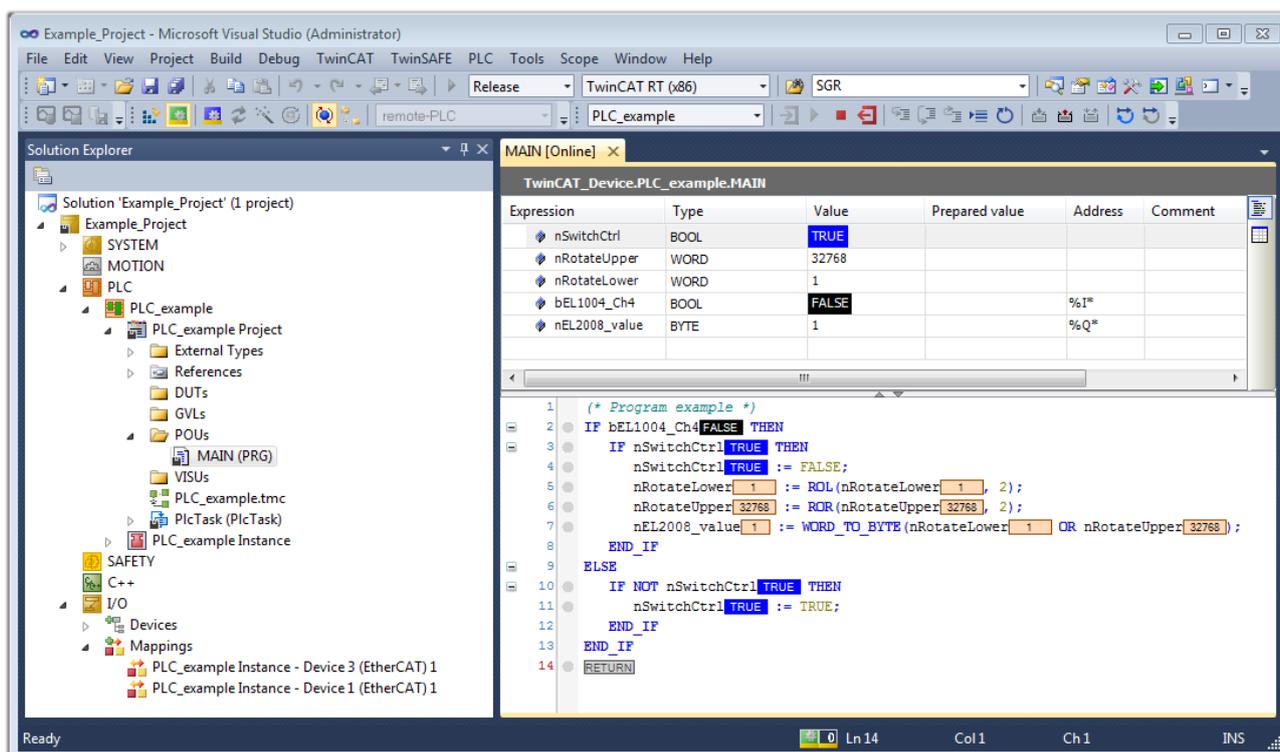


図 61: TwinCAT開発環境 (VSシェル): プログラム起動後にログイン

オペレータ制御エレメント  をクリックすると停止、 をクリックするとログアウトします(「Shift + F5」でも停止します。どちらの操作もPLCメニューから選択可能です)。

5.2 TwinCAT開発環境

オートメーションTwinCAT (The Windows Control and Automation Technology)用のソフトウェアには、以下の2つがあります。

- ・ TwinCAT 2: System Manager (コンフィグレーション) & PLC制御 (プログラミング)
- ・ TwinCAT 3: TwinCAT 2の拡張版 (共通の開発環境でプログラミングとコンフィグレーションが可能)

詳細:

- ・ **TwinCAT 2:**
 - 変数指向でI/Oデバイスをタスクに接続
 - 変数指向でタスクをタスクに接続
 - ユニットをビットレベルでサポート
 - 同期または非同期関係をサポート
 - コンシステントなデータ領域およびプロセスイメージの交換
 - NT上でのデータリンク - オープンなMicrosoft規格のプログラム (OLE、OCX、ActiveX、DCOM+など)
 - Windows NT/2000/XP/Vista、Windows 7、NT/XP Embedded、GEでのIEC 61131-3-ソフトウェア-SPS、ソフトウェア-NC、およびソフトウェア -CNCの統合
 - 一般的なすべてのフィールドバスとの相互接続
 - その他...

その他の機能:

- ・ **TwinCAT 3 (eXtended Automation):**
 - Visual-Studio®統合
 - プログラミング言語の選択
 - IEC 61131-3のオブジェクト指向の拡張をサポート
 - リアルタイムアプリケーションのプログラミング言語としてC/C++を使用
 - MATLAB®/Simulink®への接続
 - 拡張性に富んだオープンインターフェイス
 - 柔軟なランタイム環境
 - マルチコアおよび64ビットオペレーティングシステムのアクティブサポート
 - TwinCATオートメーションインターフェイスによる自動コード生成およびプロジェクト作成
 - その他...

以降のセクションでは、制御用PCシステム上でのTwinCAT開発環境のコミッショニング、および独自の制御エレメントの基本機能について説明します。

TwinCAT 2およびTwinCAT 3の詳細情報は、<http://infosys.beckhoff.com>を参照してください。

5.2.1 TwinCATリアルタイムドライバのインストール

IPCコントローラの標準イーサネットポートにリアルタイム機能を割り当てるには、このポートに対してWindowsでベッコフ リアルタイムドライバをインストールする必要があります。

これは、複数の方法で行うことができます。ここではオプションの1つについて説明します。

System Managerで、[Options] → [Show Real Time Ethernet Compatible Devices]からローカルネットワークインターフェイスの設定ダイアログを起動します。

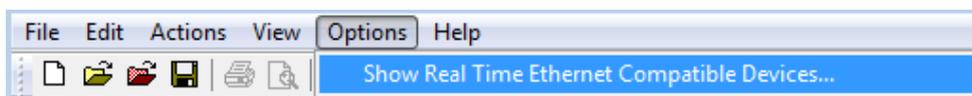


図 62: System Managerの[Options] (TwinCAT 2)

TwinCAT 3環境では、メニュー[TwinCAT]で開始する必要があります。

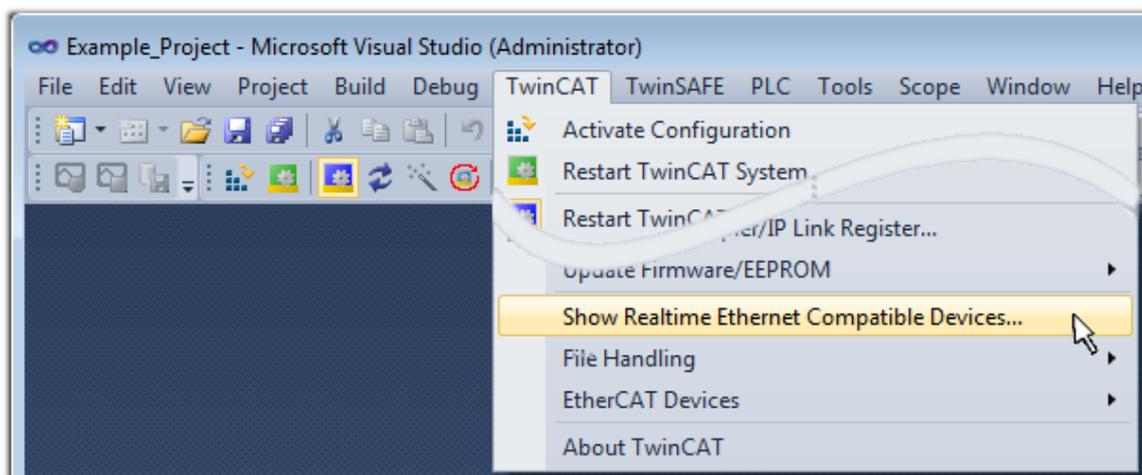


図 63: VSシェルでの起動(TwinCAT 3)

次のダイアログが表示されます。

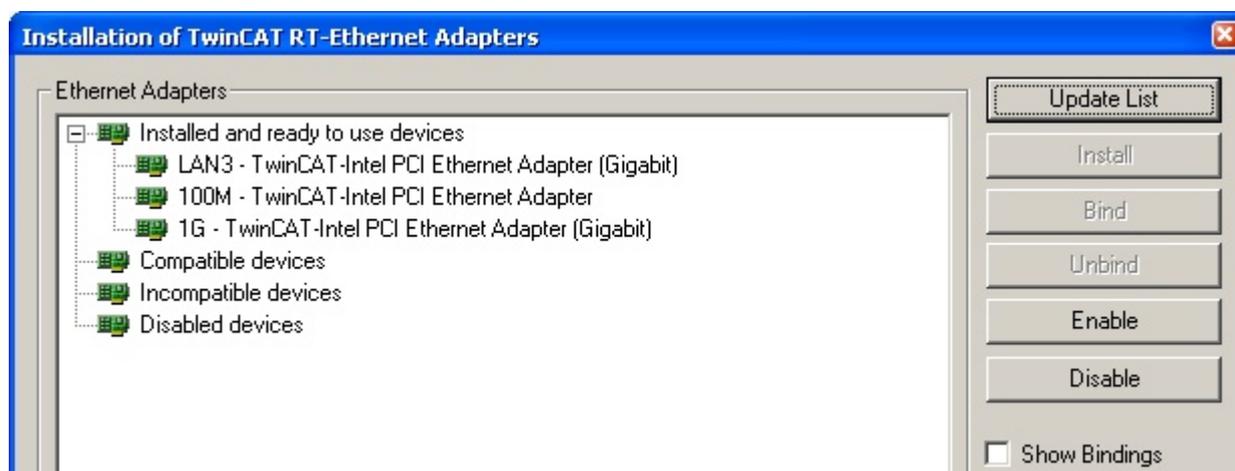


図 64: ネットワークインターフェ이스の概要

[Install] ボタンを使用して、[Compatible devices] に表示されているインターフェイスに対してドライバを割り当てられます。ドライバは、互換性のあるデバイスにのみインストールしてください。

署名のないドライバに関するWindowsの警告は、無視することができます。

または、チャプタ「オフライン構成の作成」、セクション「EtherCATデバイスの作成」 [▶ 66] の説明のとおり、EtherCATデバイスを最初に挿入し、互換性のあるイーサネットポートをEtherCATプロパティで確認することができます([Adapter] タブ、[Compatible Devices...] ボタン)。



図 65: EtherCATデバイスのプロパティ (TwinCAT 2): タブ[Adapter]の[Compatible Devices]をクリック

TwinCAT 3: [I/O]のSolution Explorer内の[Device .. (EtherCAT)]をダブルクリックすると、EtherCATデバイスのプロパティを開けます。



インストール後、ネットワークインターフェイスのWindowsの概要内に、有効になったドライバが表示されます (Windowsの[スタート] → [システムのプロパティ] → [ネットワーク])。

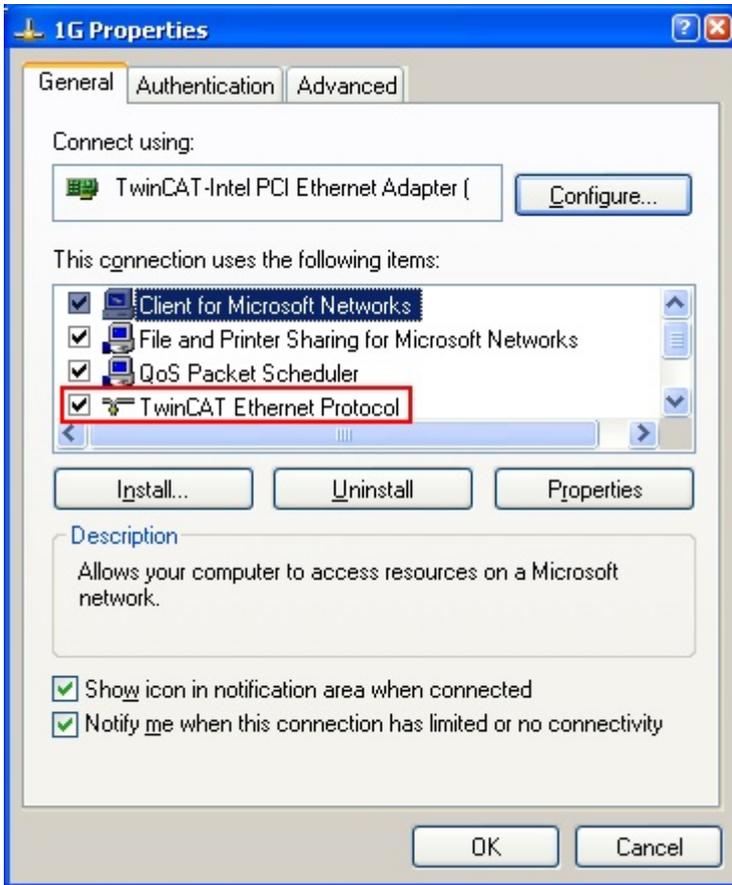


図 66: ネットワークインターフェイスのWindowsプロパティ

以下の図では、ドライバが正しく設定されています。



図 67: イーサネットポートに対する正しいドライバ設定の例

以下のような間違った設定を行うと正しく動作しません。

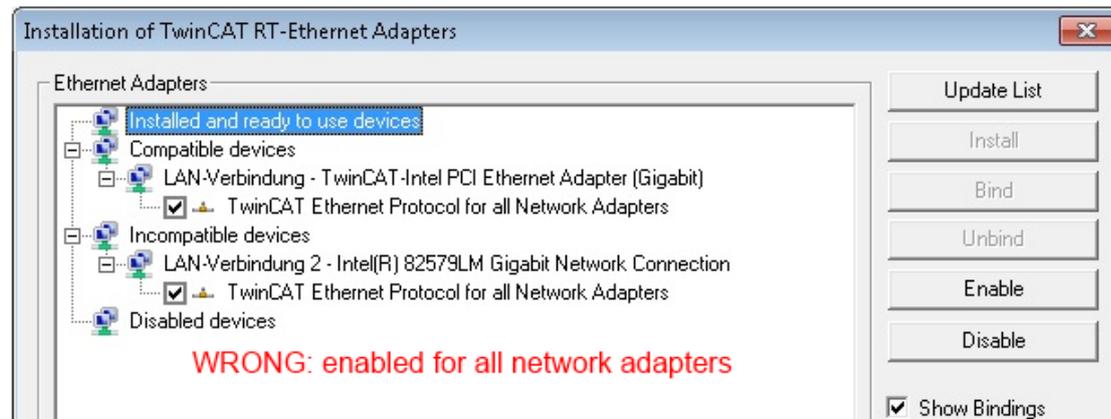
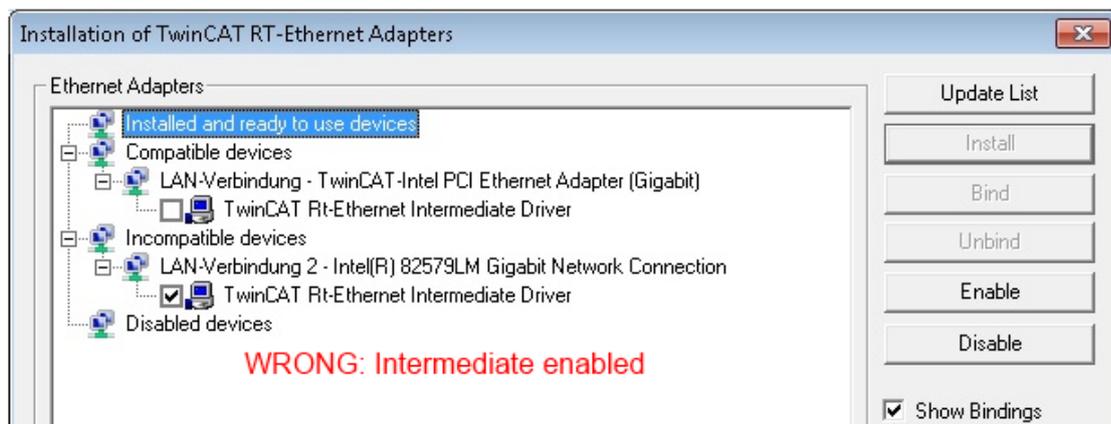
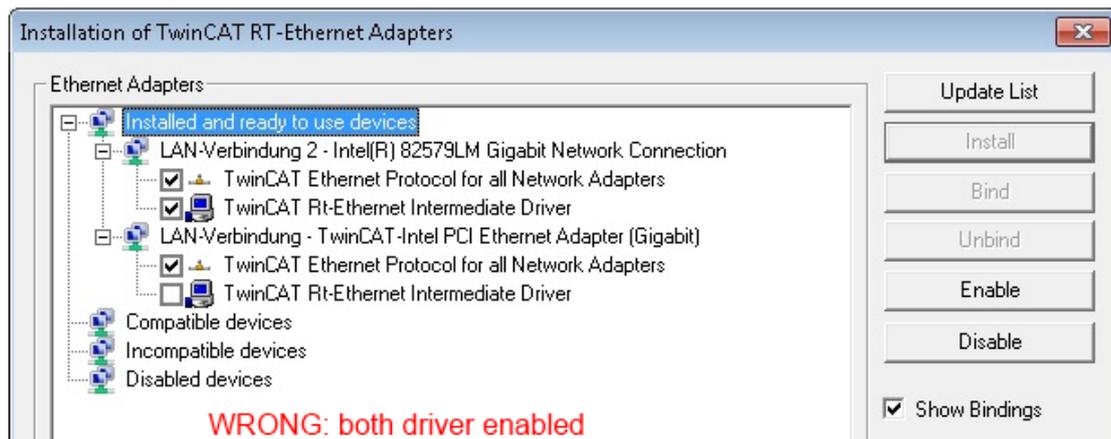


図 68: イーサネットポートに対する誤ったドライバ設定

使用するポートのIPアドレス

● IPアドレス/DHCP

i 通常、EtherCATデバイスとして構成されるイーサネットポートは、一般的なIPパケットを転送しないのでIPアドレスは必要ありません。しかし、特にEL6601や同様のデバイスを使用する場合は、[インターネットプロトコルTCP/IP]ドライバ設定でこのポートに対して固定IPアドレスを指定し、DHCPを無効にすることを推奨します。これにより、DHCPサーバが存在しなくてもイーサネットポート自身にデフォルトのIPアドレスを割り当てることで、DHCPクライアントに関連する遅延を回避できます。適切なアドレス空間は192.168.x.xなどです。

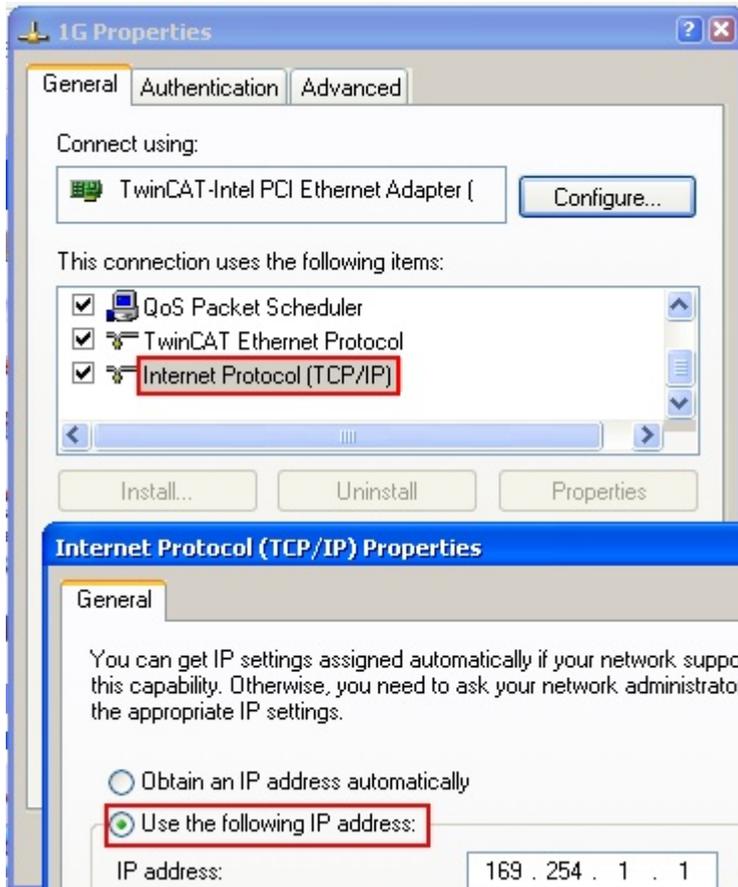


図 69: イーサネットポートのTCP/IP設定

5.2.2 ESIデバイス記述ファイルに関する注記

最新のESIデバイス記述ファイルのインストール

TwinCAT EtherCATマスタ/System Managerは、オンラインまたはオフラインモードでコンフィグレーションを生成するために、使用するデバイスの記述ファイルが必要とします。デバイス記述ファイルは、いわゆるESI (EtherCAT Slave Information) ファイルにXML形式で含まれています。これらのファイルは各メーカーに要望し、ダウンロードすることが可能です。*.xmlファイルには、複数のデバイスの説明が含まれる場合があります。

ベッコフEtherCATデバイスのESIファイルは、[ベッコフ ウェブサイト](#)から入手可能です。

ESIファイルは、TwinCATのインストールディレクトリに保存する必要があります。

デフォルト設定:

- ・ **TwinCAT 2:** C:\¥TwinCAT¥IO¥EtherCAT
- ・ **TwinCAT 3:** C:\¥TwinCAT¥3.1¥Config¥Io¥EtherCAT

[System Manager]ウィンドウが最後に開いた後にESIファイルが変更されている場合、[System Manager]ウィンドウを新しく開くとこのファイルが一度読み込まれます。

TwinCATのインストールには、TwinCATのビルド作成時に最新のベッコフESIファイル群が含まれています。

TwinCAT 2.11/TwinCAT 3以降では、プログラミングPCがインターネットに接続されている場合、System ManagerからESIディレクトリを更新できます。更新は、以下のメニューから行います。

- ・ **TwinCAT 2:** [Option] → [Update EtherCAT Device Descriptions]
- ・ **TwinCAT 3:** [TwinCAT] → [EtherCAT Devices] → [Update Device Descriptions (via ETG Website) ...]

ESIディレクトリ更新のために、TwinCAT ESI Updaterが用意されています。

ESI

.xmlファイルは、ESI XMLファイルの構造を記述した.xsdファイルに関連付けられています。このため、ESIデバイス記述ファイルを更新する場合は、両方のファイルタイプを更新する必要があります。

デバイスの区別

EtherCATデバイス/スレーブは、完全なデバイス識別子を構成する4つのプロパティによって区別されます。例えば、デバイス識別子EL2521-0025-1018は以下で構成されています。

- ・ ファミリーキー「EL」
- ・ 名前「2521」
- ・ タイプ「0025」
- ・ リビジョン「1018」

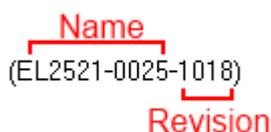


図 70: 識別子の構造

名前+タイプで構成されるオーダー識別子(ここではEL2521-0010)は、デバイス機能を示します。技術的な更新を示すリビジョンは、ベッコフが管理しています。原則として、取扱説明書などに記載のない限り、上位リビジョンのデバイスで下位リビジョンのデバイスを置換できます。各リビジョンには、専用のESIファイルが用意されています。その他の注記 [▶ 7] も参照してください。

オンラインディスクリプション

実際のデバイスのスキャン(セクション「オンラインセットアップ」を参照)によってEtherCATコンフィグレーションがオンラインで作成され、スレーブ用のESIファイルが検出されない場合、System Managerはデバイスに保存されている記述ファイルを使用するかどうかを確認します。スレーブとの同期および非同期通信を設定するために、System Managerはどのような場合であってもこの情報を必要とします。

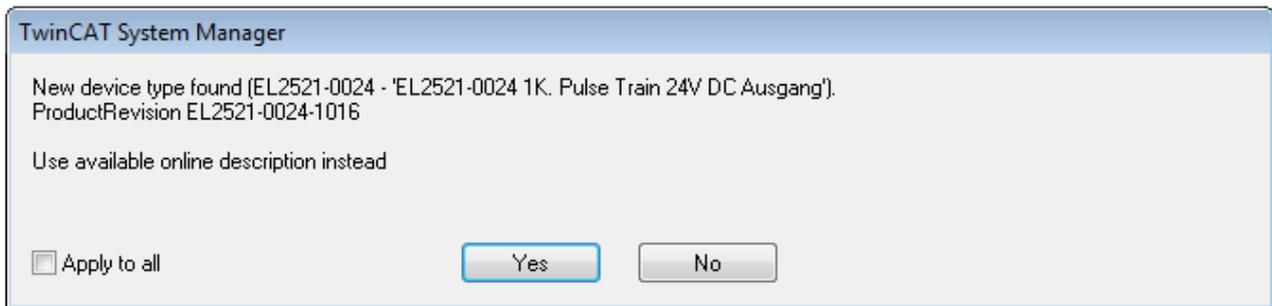


図 71: オンラインディスクリプション情報ウィンドウ (TwinCAT 2)

TwinCAT 3でも同様のウィンドウが表示され、ここからオンライン更新ができます。

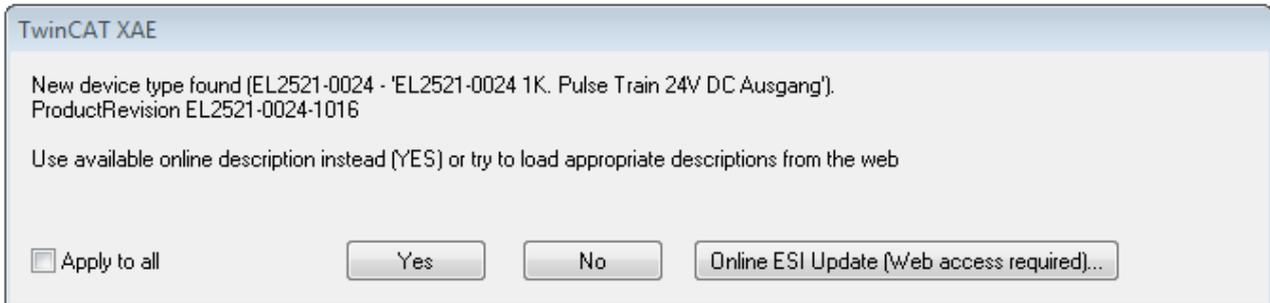


図 72: オンラインディスクリプション情報ウィンドウ (TwinCAT 3)

デバイスメーカーから必要なESIが入手可能な場合は、[Yes]と回答せず、ESIをメーカーにリクエストすることをお勧めします。XML/XSDファイルのインストール後、設定処理を再度行う必要があります。

注記

スキャンによる「通常」コンフィグレーションの変更

- ✓ スキャンすることによってTwinCATがまだ認識していないデバイスを検出した場合、以下の2つの場合で操作が異なります。ここでは、リビジョン1019のEL2521-0000を例として説明します。
 - a) EL2521-0000デバイスのリビジョン1019用のESIも、それ以前のリビジョン用のESIも存在しない場合。この場合、メーカー(この例ではベッコフ)にESIをリクエストする必要があります。
 - b) EL2521-0000デバイスの以前のリビジョン(1018や1017など)用のESIしか存在しない場合。この場合、先にスペアパーツの在庫を確認し、コンフィグレーションに新しいリビジョンを統合しても問題がないかを判断するためのインハウスチェックを行う必要があります。新しい上位のリビジョンには、新機能が付随することが多くあります。新機能を使用しない場合は、コンフィグレーションが以前の1018のリビジョンに対応していれば運用を継続できます。これは、ベッコフの互換性に関するルールでも規定されています。

チャプタ「ベッコフEtherCAT IOコンポーネントの使用に関する一般的な注記」、および手動コンフィグレーションの場合はチャプタ「オフラインでのコンフィグレーションの作成 [▶ 66]」を参照してください。

意図せずオンラインディスクリプションが使用されている場合、System ManagerはEtherCATスレーブのEEPROMからデバイス記述ファイルのコピーを読み込みます。コンプレックススレーブでは、EEPROMのサイズが完全なESI情報に対して不十分なことがあり、この場合はコンフィグレータ内でESIが不完全な状態になります。このため、コンプレックススレーブではオフラインESIファイルを優先的に使用することを推奨します。

System Managerはオンラインで記録するデバイスの説明に対して、オンラインで読み込まれたすべてのESIの内容を含むファイル「OnlineDescription0000...xml」をESIディレクトリ内に新規作成します。

OnlineDescriptionCache000000002.xml

図 73: System Managerによって作成されたOnlineDescription.xmlファイル

後でコンフィグレーションにスレーブを手動で追加する場合のために、選択リスト内のオンライン作成されたスレーブの前に記号「>」が表示されます(図「SL2521のオンラインで記録されたESI表示の例」を参照)。

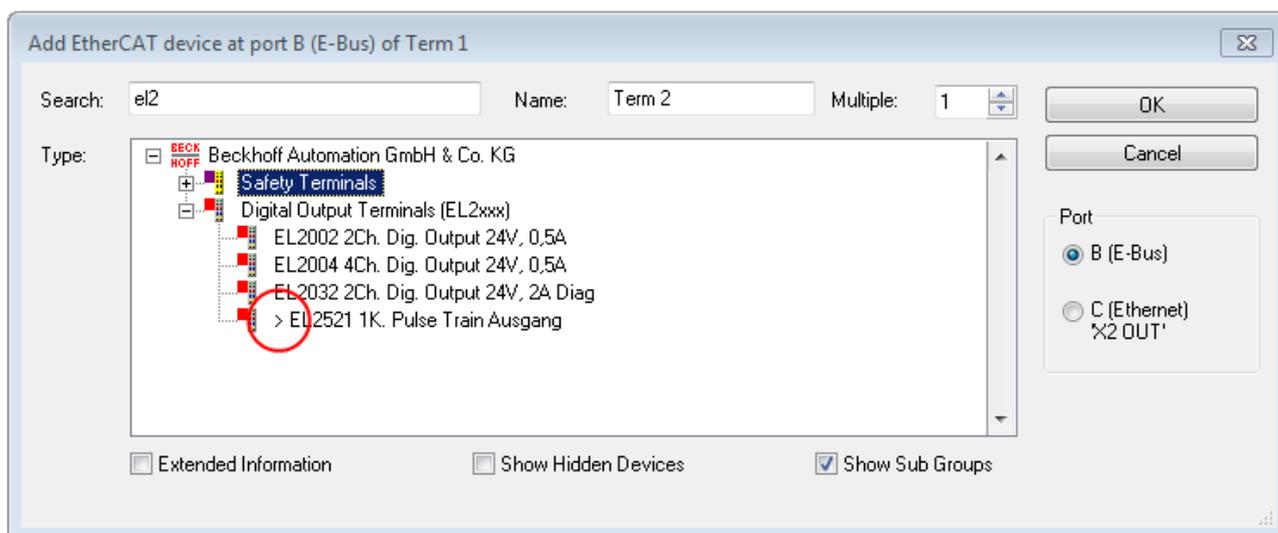


図 74: EL2521のオンラインで記録されたESI表示の例

このようなESIファイルが使用されていて、後からメーカーのファイルが利用可能になった場合、OnlineDescription.xmlファイルは以下の手順で削除する必要があります。

- ・ すべてのSystem Managerウィンドウを閉じる。
- ・ TwinCATをConfigモードで再起動する。
- ・ 「OnlineDescription0000...xml」を削除する。
- ・ TwinCAT System Managerを再起動する。

この操作を行うと、このファイルは表示されなくなります。表示を更新する必要がある場合は、<F5>を押します。

● TwinCAT 3.xのオンラインディスクリプション

i 前述の「OnlineDescription0000...xml」に加えて、Windows 7などでは新しく検出されたデバイスのいわゆるEtherCATキャッシュがTwinCAT 3.xによって作成されます。

`C:\User\[USERNAME]\AppData\Roaming\Beckhoff\TwinCAT3\Components\Base\EtherCATCache.xml`

(OSの言語設定に注意してください。)

このファイルも削除する必要があります。

問題のあるESIファイル

ESIファイルに問題があり、System Managerがこのファイルを読み込めない場合、System Managerは情報ウィンドウを表示します。

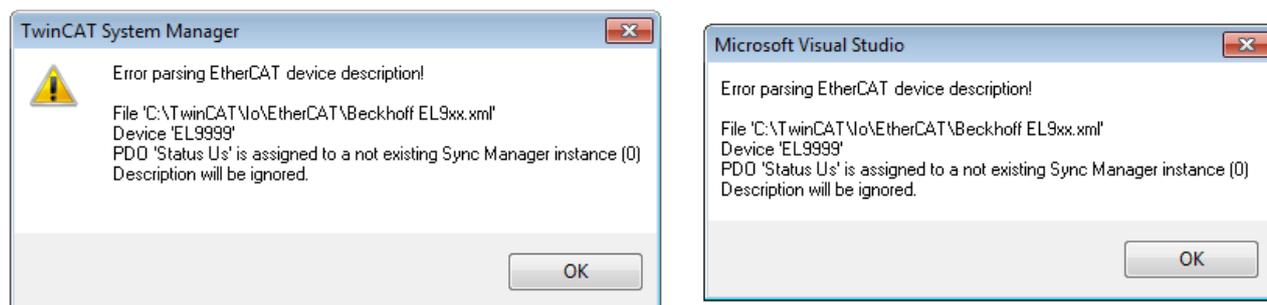


図 75: ESIファイルに問題があった場合に表示される情報ウィンドウ(左: TwinCAT 2、右: TwinCAT 3)

考えられる理由:

- ・ *.xmlの構造が、関連する*.xsdファイルに対応していない → スキームをチェック
- ・ コンテンツをデバイス記述ファイルに変換できない → ファイルの製造者に問い合わせ

5.2.3 オフラインでのコンフィグレーションの作成

EtherCATデバイスの作成

空のSystem Managerウィンドウ内でEtherCATデバイスを作成します。

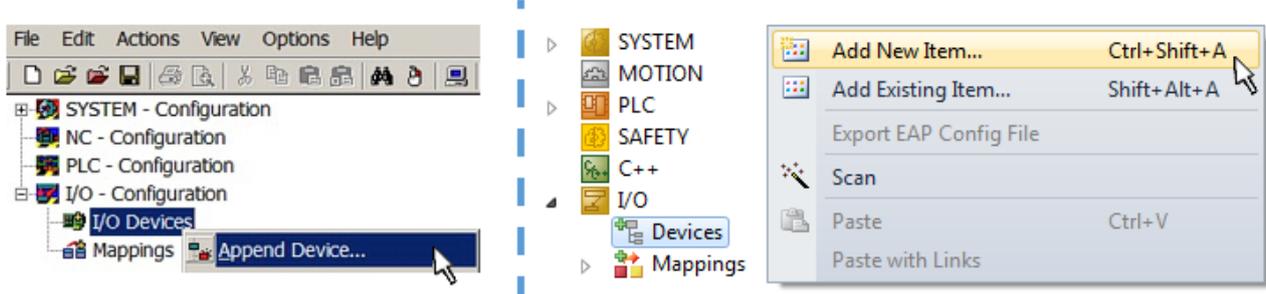


図 76: EtherCATデバイスの追加(左: TwinCAT 2、右: TwinCAT 3)

EtherCATスレーブを使用するEtherCAT I/Oアプリケーションに対して、タイプ[EtherCAT]を選択します。現在のパブリッシャ/サブスクリバサービスとEL6601/EL6614ターミナルの組合せに対して、[EtherCAT Automation Protocol via EL6601]を選択します。

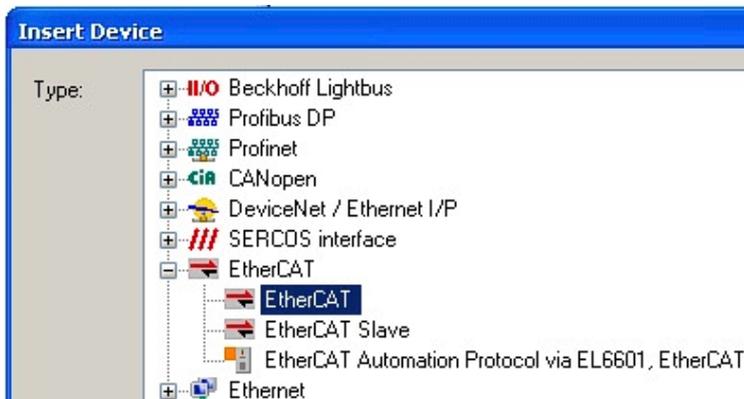


図 77: EtherCAT接続の選択(TwinCAT 2.11、TwinCAT 3)

次に、ランタイムシステム内でこの仮想デバイスに対して実際のイーサネットポートを割り当てます。

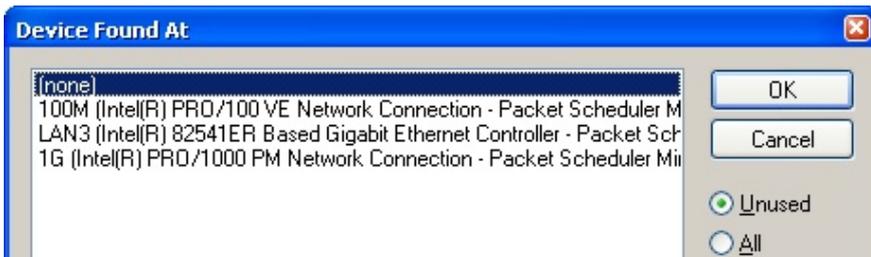


図 78: イーサネットポートの選択

EtherCATデバイスの作成時に、この確認ダイアログが自動的に表示される場合があります。プロパティダイアログで後から割り当てを設定/変更することも可能です。図. 「EtherCATデバイスのプロパティ(TwinCAT 2)」を参照してください。

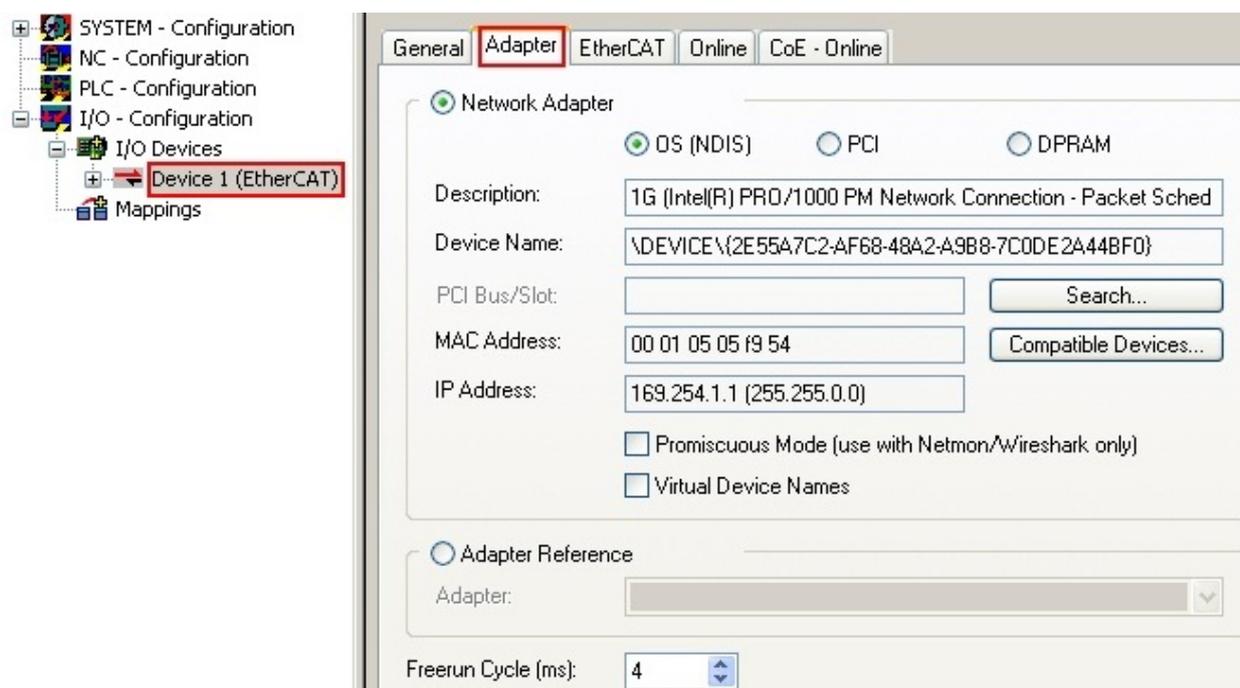


図 79: EtherCATデバイスのプロパティ (TwinCAT 2)

TwinCAT 3: [I/O]のSolution Explorer内の[Device ... (EtherCAT)]をダブルクリックすると、EtherCATデバイスのプロパティを開けます。



イーサネットポートの選択

イーサネットポートは、EtherCATデバイスに対してTwinCATリアルタイムドライバがインストールされているポートのみ選択が可能です。この操作は、各ポートに対して個別に行う必要があります。該当するインストールの説明 [▶ 58] を参照してください。

EtherCATスレーブの追加

デバイスをさらに追加するには、コンフィグレーションツリー内でデバイスを右クリックします。

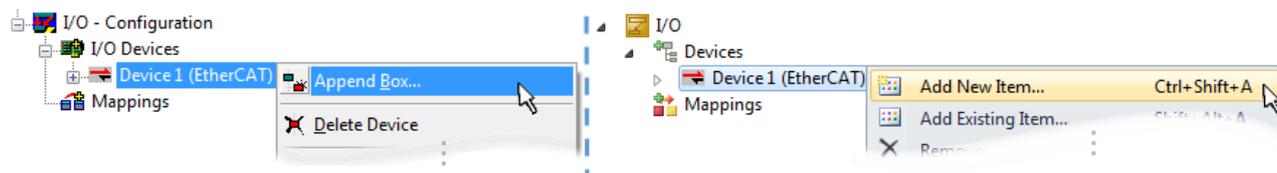


図 80: EtherCATデバイスの追加(左: TwinCAT 2、右: TwinCAT 3)

新しいデバイスを選択するためのダイアログが開きます。ESIファイルが用意されているデバイスしか表示されません。

既に選択されているデバイスに追加可能なデバイスのみ、選択が可能です。このため、このポートで使用可能な物理層も表示されます(図。「新しいEtherCATデバイスの選択ダイアログ」、A)。PHY転送に対応したケーブルベースのファーストイーサネット物理層の場合、図。「新しいEtherCATデバイスの選択ダイアログ」のように、ケーブルベースのデバイスしか使用できません。前のデバイスに複数の空きポートがある場合(EK1122やEK1100など)、必要なポートを右側(A)で選択できます。

物理層の概要

- ・ 「イーサネット」： ケーブルベースの100BASE-TX: RJ45/M8/M12コネクタ対応のEKカプラ、EPボックス、デバイス
- ・ 「Eバス」： LVDS「ターミナルバス」「EJモジュール」： EL/ESターミナル、各種モジュール

検索フィールドを使用すると、特定のデバイスを簡単に見つけられます(TwinCAT 2.11以降、またはTwinCAT 3)。

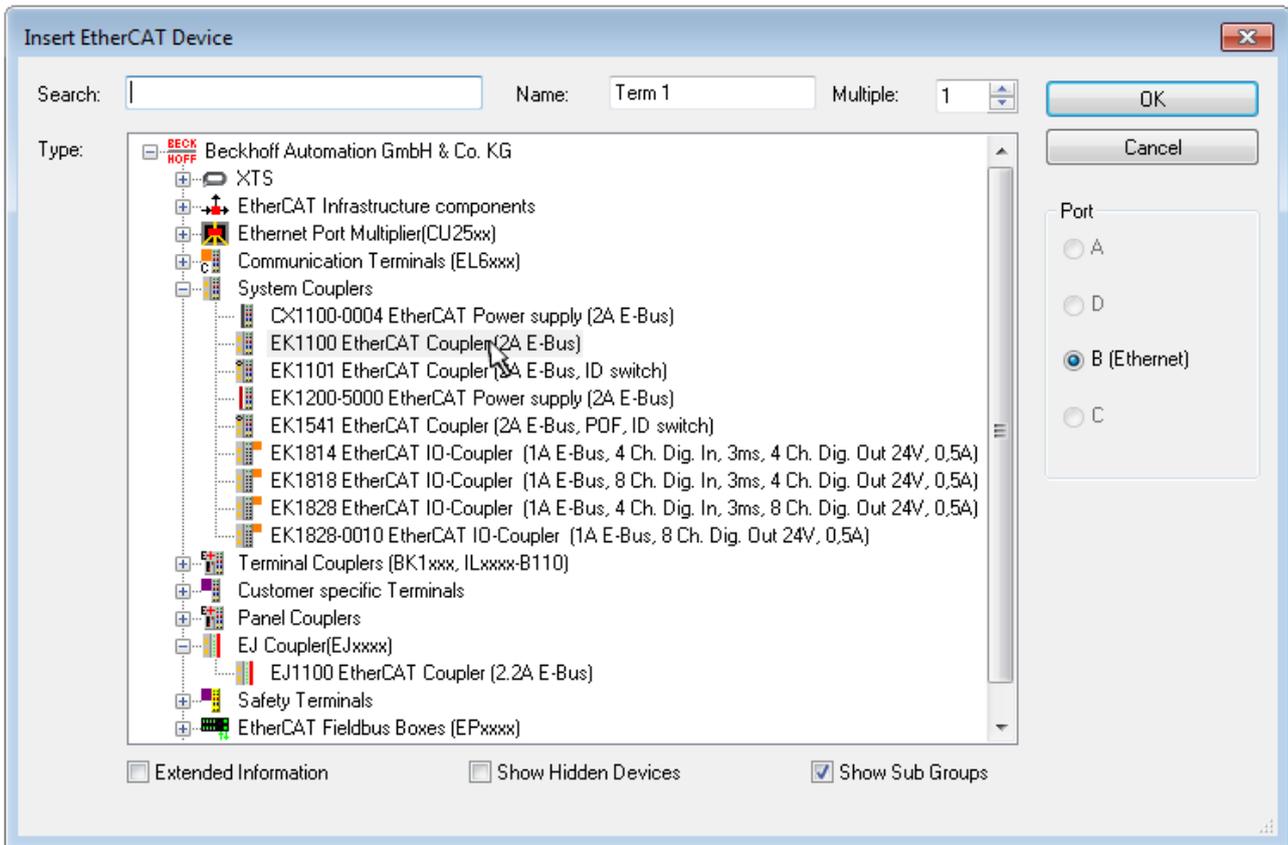


図 81: EtherCATデバイスの選択ダイアログ

デフォルトでは、名前/デバイスタイプのみが選択基準として使用されます。[Extended Information]としてリビジョンを表示すると、デバイスの特定のリビジョンを選択できます。

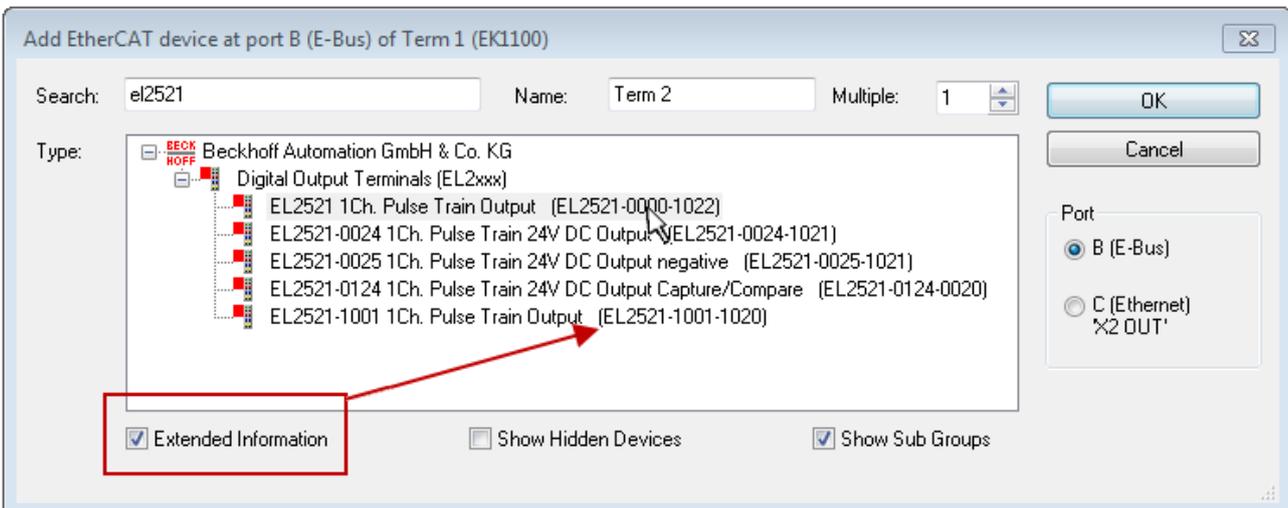


図 82: デバイスのリビジョンの表示

多くの場合、技術的な進歩などによるアップデートや機能上の理由で、複数のデバイスリビジョンが作成されます。ベッコフデバイスの選択画面では、分かりやすさを考慮して(図. 「新しいEtherCATデバイスの選択ダイアログ」を参照)、最新の(上位の)リビジョンと製品の最新の状態のみが表示されます。ESIファイルとして用意されているデバイスリビジョンをすべて表示するには、[Show Hidden Devices]チェックボックスにチェックを入れます(図. 「以前のリビジョンの表示」を参照)。

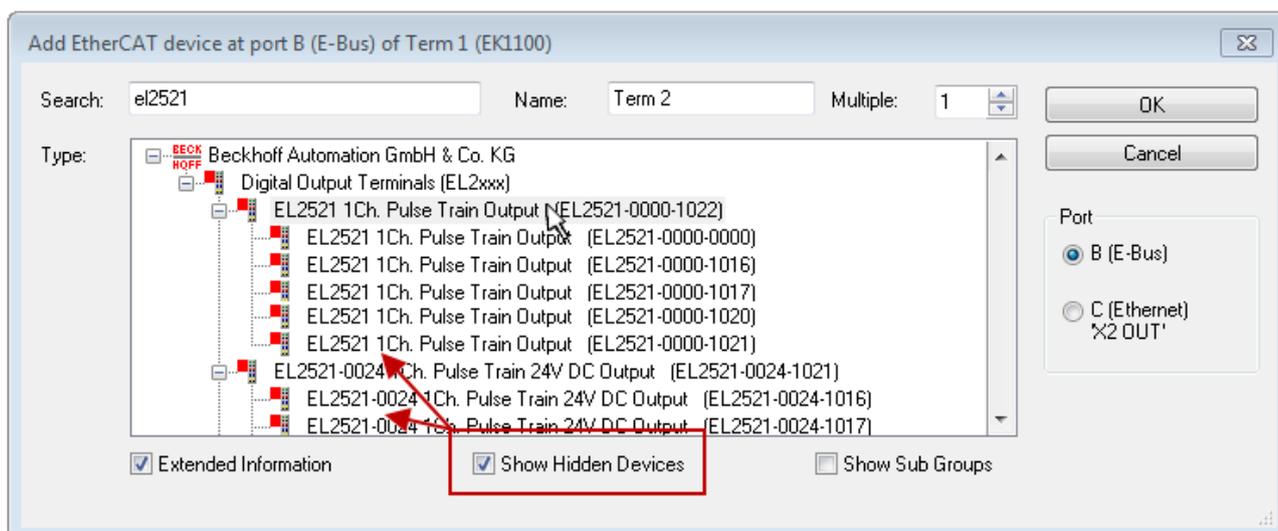


図 83: 以前のリビジョンの表示

● リビジョンに基づいたデバイス選択、互換性

ESIファイルは、プロセスイメージ、マスタとスレーブ/デバイス間の通信タイプ、および適用可能な場合はデバイス機能も定義します。物理デバイス(使用可能な場合はファームウェア)は、マスタの通信確認ダイアログ/設定をサポートしている必要があります。これは後方互換であり、EtherCATマスタが新しいデバイス(上位リビジョン)を古いリビジョンとして扱う場合でも、そのデバイスがサポートされます。ベッコフEtherCATターミナル/ボックス/EJモジュールは、以下の互換性を前提としています。

システム内のデバイスリビジョン ≧ コンフィグレーション内のデバイスリビジョン

これにより、コンフィグレーションを変更せずに、後でデバイスを交換することが可能です(ドライブに対して異なる指定が可能)。

例:

コンフィグレーションでEL2521-0025-1018が指定されている場合、実際にはEL2521-0025-1018以降(-1019、-1020)を使用できます。

Name
(EL2521-0025-1018)
Revision

図 84: ターミナルの名前/リビジョン

TwinCATシステム内で最新のESIファイルが使用できる場合は、選択ダイアログで候補として表示される最新のリビジョンが、ベッコフの現行製品に適合します。実際のアプリケーションで最新のベッコフデバイスを使用する場合は、コンフィグレーションを新規作成する際に最新のデバイスリビジョンを使用することを推奨します。古いリビジョンは、在庫していた古いデバイスをアプリケーション内で使用する場合のみ使用してください。

この場合、デバイスのプロセスイメージがコンフィグレーションツリーに表示され、以下のパラメータ設定が可能になります: タスクとのリンク、CoE/DC設定、プラグイン定義、スタートアップ設定など。

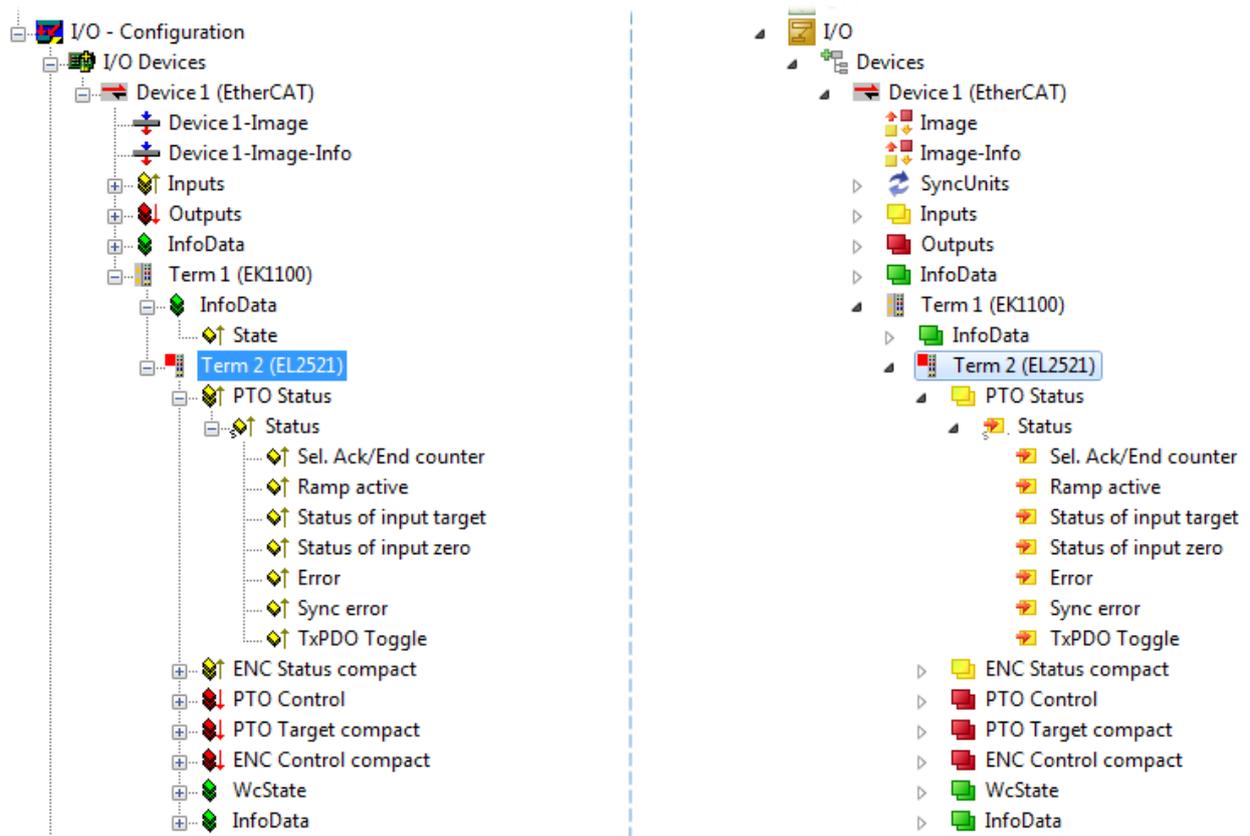


図 85: TwinCATツリー内のEtherCATターミナル(左: TwinCAT 2、右: TwinCAT 3)

5.2.4 オンラインでのコンフィグレーションの作成

EtherCATデバイスの検出/スキャン

TwinCATシステムがConfigモードの場合、オンラインデバイス検索を使用できます。これは、情報バーの右下に表示されるアイコンで示されます。

- ・ TwinCAT 2では、System Managerウィンドウ内に青で[Config Mode]と表示されます: 。
- ・ TwinCAT 3では、開発環境のユーザインターフェイス内にアイコン  が表示されます。

TwinCATは以下の方法でConfigモードに設定できます。

- ・ TwinCAT 2: メニューバーの  を選択、または[Actions] → [Set/Reset TwinCAT to Config Mode...]
- ・ TwinCAT 3: メニューバーの  を選択、または[TwinCAT] → [Restart TwinCAT (Config Mode)]

● Configモードでのオンラインスキャン

i RUNモード(製造オペレーション)では、オンライン検索を使用できません。TwinCATプログラミングシステムとTwinCATターゲットシステムの違いに注意してください。

Windowsのタスクバー内のTwinCAT 2アイコン()またはTwinCAT 3アイコン()が、ローカルIPCのTwinCATモードを常時表示します。一方で、TwinCAT 2のSystem Managerウィンドウ、またはTwinCAT 3のユーザインターフェイスはターゲットシステムの状態を示します。



図 86: ローカルシステムとターゲットシステムの違い(左: TwinCAT 2、右: TwinCAT 3)

コンフィグレーションツリーの[I/O Devices]を右クリックすると、検索ダイアログが開きます。

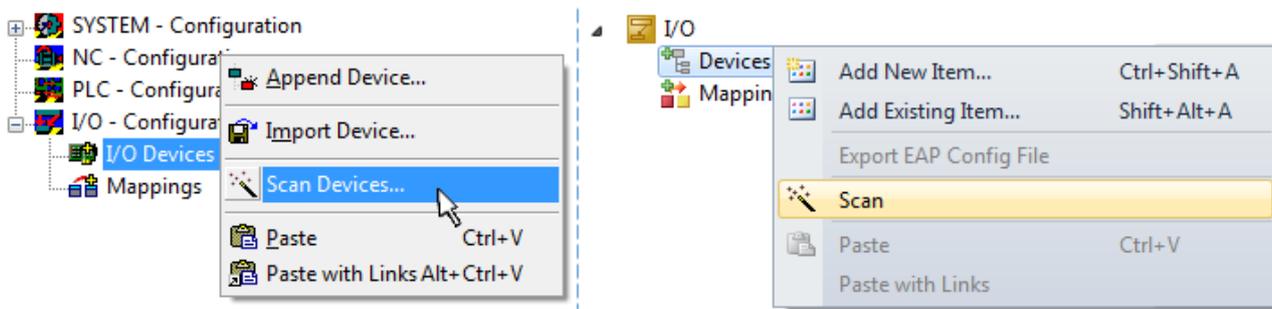


図 87: デバイスのスキャン(左: TwinCAT 2、右: TwinCAT 3)

このスキャンモードでは、EtherCATデバイス(またはEtherCATデバイスとして使用可能なイーサネットポート)だけでなく、NOVRAM、フィールドバスカード、SMBなども検索されます。ただし、自動的にすべてのデバイスが検出されるわけではありません。

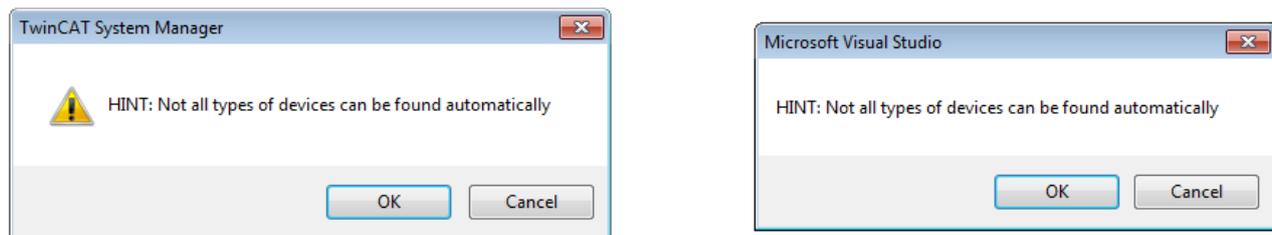


図 88: 自動デバイススキャンに関する注意(左: TwinCAT 2、右: TwinCAT 3)

イーサネットポートとインストール済みのTwinCATリアルタイムドライバが、「RT Ethernet」デバイスとして表示されます。テストのために、EtherCATフレームがこれらのポートに送信されます。EtherCATスレーブが接続されていることを示す応答をスキャンエージェントが検出すると、ポートがすぐに「EtherCAT Device」として表示されます。

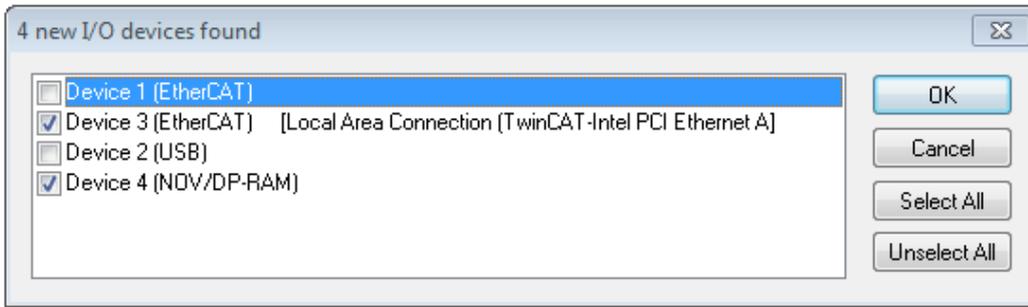


図 89: 検出されたイーサネットデバイス

各チェックボックスを使用して、デバイスを選択できます(図。「検出されたイーサネットデバイス」では「Device 3」と「Device 4」を選択)。**[OK]**で確定後、選択されたすべてのデバイスに対するデバイススキキャンが提案されます。図。「EtherCATデバイスの自動作成後のスキャン開始確認ダイアログ」を参照してください。

● イーサネットポートの選択

i イーサネットポートは、EtherCATデバイスに対してTwinCATリアルタイムドライバがインストールされているポートのみ選択が可能です。この操作は、各ポートに対して個別に行う必要があります。該当するインストールの説明 [▶ 58] を参照してください。

EtherCATデバイスの検出/スキャン

● オンラインスキャン機能

i スキャン中、マスタはEtherCATスレーブの識別情報をスレーブ内のEEPROMから照会します。タイプの判別には、名前とリビジョンが使用されます。対応するデバイスは保存されているESIデータ内に存在し、そこで定義されているデフォルト状態でコンフィグレーションツリーに統合されます。

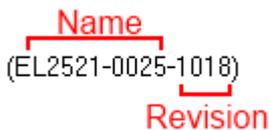


図 90: デフォルト状態の例

注記

マシンの量産時における実際のスレーブスキャン

スキャン機能の使用には注意が必要です。スキャン機能は、コミッショニングのベースとしての初期コンフィグレーションを手早く作成するための実践的なツールです。ただし、プラントでのマシンの量産や再生産では、定義されている初期コンフィグレーションとの比較 [▶ 75] で必要な場合を除き、この機能をコンフィグレーションの作成に使用するべきではありません。理由: ベッコフは製品をメンテナンスするために、納品済み製品のリビジョンバージョンを上げることがあり、スキャンすることによってデバイスリストに基づいた同一(同一の機械構造)のコンフィグレーションが作成されます。ただし、各デバイスのリビジョンが、初期コンフィグレーションとは異なる可能性があります。

例:

会社Aが、マシンBのプロトタイプを製造します。このマシンは、後で量産します。これを行うために、プロトタイプを製造し、TwinCATでI/Oデバイスのスキャンを実行して、初期コンフィグレーション「B.tsm」を作成します。このマシンには、リビジョン1018のEL2521-0025 EtherCATターミナルを搭載します。このターミナルは、以下の方法でTwinCATコンフィグレーションに組み込みます。

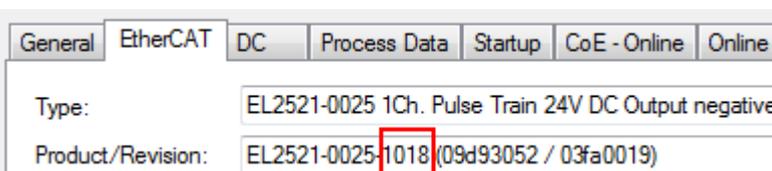


図 91: リビジョン1018のEtherCATターミナルのインストール

プロトタイプテストフェーズ中には、プログラマ/コミッショニングエンジニアがこのターミナルの機能およびプロパティをテストし、必要に応じてPLC「B.pro」またはNCからアドレス指定します(TwinCAT 3ソリューションファイルの場合も同様です)。

これでプロトタイプの開発は完了し、マシンBの量産が開始されます。量産時、ベッコフは継続的にEL2521-0025-0018を提供します。必ずマシンの量産部門のコミッショニングエンジニアがスキャンを実行すれば、全マシンに対してコンフィグレーションBの内容は同一になります。会社Aは、EL2521-0025-1018ターミナルを搭載した量産型マシンに対応可能な保守部材のグローバル調達網を確保します。

しばらくして、ベッコフはEL2521-0025を新機能Cで拡張します。このため、FWが変更されます。変更したFWは、上位となるFWバージョンと新しいリビジョン-1019で識別可能です。この場合でも、以前のバージョンの機能やインターフェイスはそのままサポートされます。このため、「B.tsm」、さらには「B.pro」でさえも変更不要です。マシンは「B.tsm」および「B.pro」を使用して継続的に量産が可能です。生産されたマシンをチェックすることを目的として、初期コンフィグレーション「b.tsm」との比較スキャン [▶ 75] が行われる可能性があります。

ただし、マシンの量産部門が「B.tsm」は使用せず、生産用のコンフィグレーション作成のためにスキャンを実行すると、自動的にリビジョン-1019が検出され、コンフィグレーションに組み込まれます。

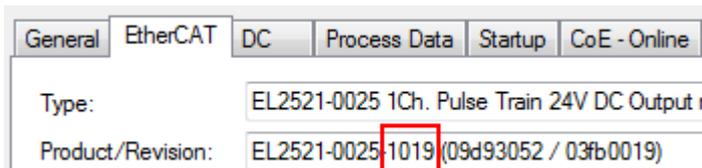


図 92: リビジョン-1019のEtherCATターミナルの検出

通常、コミッショニングエンジニアがこの処理について意識することはありません。実質的には新しいコンフィグレーションが作成されるため、TwinCATが何か信号を送信することはありません。ただし、互換性のルールに基づくと、これはEL2521-0025-1018がスペアパーツとしてこのマシンに組み込まれてはならないことを意味します(実際にはほとんどの場合で1018も動作します)。

加えて、生産に付随する開発の過程で、会社AがEL2521-0025-1019の新機能(アナログフィルタの改良や、診断用のプロセスデータの追加など)に着目し、社内での検討なしでこれらの新機能を使用することもあり得ます。この場合は、この方法で作成された新しいコンフィグレーション「B2.tsm」では、以前の保守用デバイスの在庫は使用できなくなります。マシンが既に量産体制に入っている場合は、スキャンは定義されている初期コンフィグレーションと比較するための情報収集のみを目的として実行するべきであり、変更を行う際には十分な注意が必要です。

コンフィグレーション内にEtherCATデバイスが作成されている場合(手動またはスキャンを使用)、デバイス/スレーブのI/Oフィールドをスキャンできます。



図 93: EtherCATデバイスの自動作成後のスキャン開始確認ダイアログ(左: TwinCAT 2、右: TwinCAT 3)

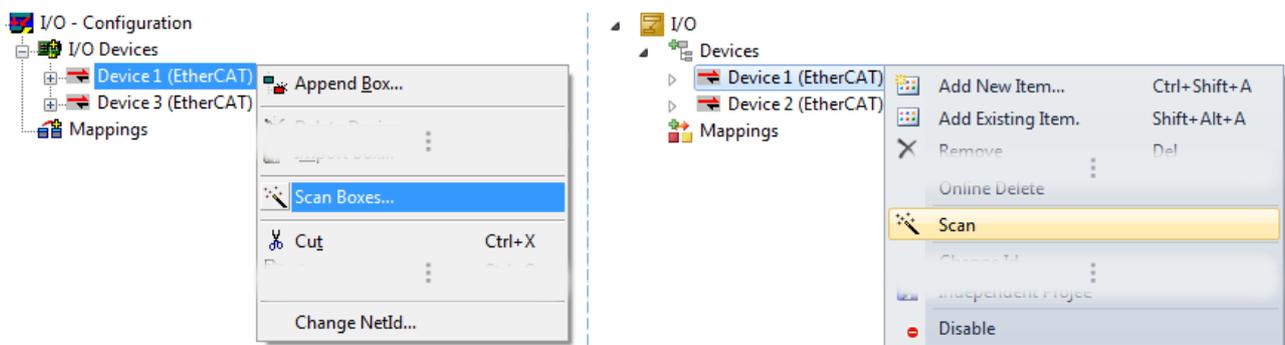


図 94: 指定したEtherCATデバイスに対するデバイススキャンの手動操作(左: TwinCAT 2、右: TwinCAT 3)

System Manager (TwinCAT 2) またはユーザインターフェイス (TwinCAT 3) 内では、下部のステータスバーにある進捗バーでスキャン処理をモニタリングできます。

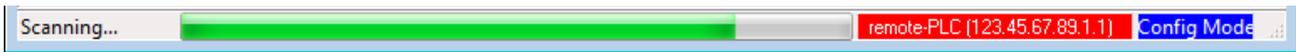


図 95: TwinCAT 2によるスキャンの進捗の例

コンフィグレーションが確立すると、オンライン状態に切り替わります (OPERATIONAL)。



図 96: Config/FreeRun確認ダイアログ(左: TwinCAT 2、右: TwinCAT 3)

Config/FreeRunモードでは、System Managerの表示が青と赤で切り替わり、アクティブなタスク (NC、PLC) がなくてもEtherCATデバイスが4 msのアイドルサイクルタイム(デフォルト設定)で動作を継続します。



図 97: 右下のステータスバーでの「FreeRun」と「Config Mode」の交互表示



図 98: ボタンを使用してTwinCATをこの状態に切り替えることも可能(左: TwinCAT 2、右: TwinCAT 3)

これで、EtherCATシステムは図. 「オンラインディスプレイの例」に示されるような、機能的なサイクリック状態となります。

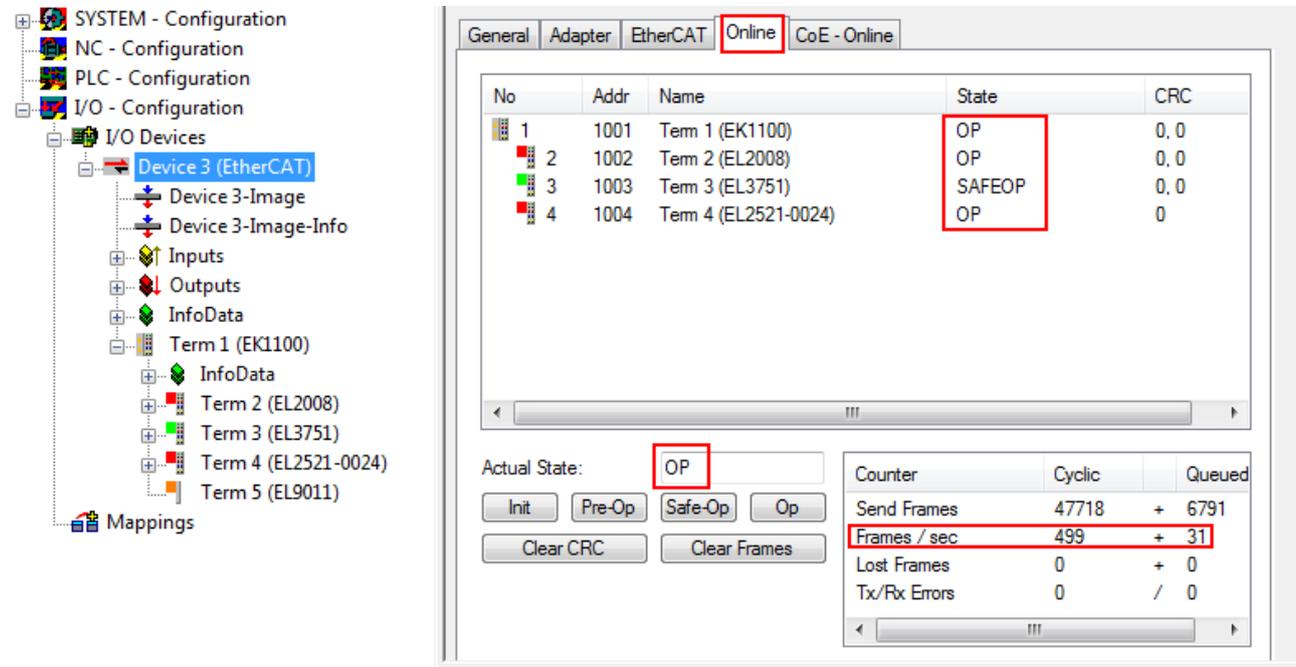


図 99: オンライン表示の例

以下に注意してください。

- ・ すべてのスレーブがOP状態であること
- ・ EtherCATマスタの[Actual State]がOPであること
- ・ 送信フレーム数を考慮した上で、[Frames/sec]とサイクルタイムが一致していること
- ・ 過度な[Lost Frames]やCRCエラーが発生しないこと

これでコンフィグレーションは完了です。コンフィグレーションは、「[手動での手順 \[▶ 66\]](#)」の記載にしたがって変更できます。

トラブルシューティング

スキャン中は、さまざまな事象が発生する可能性があります。

- ・ 使用可能なESI XMLファイルが存在しないEtherCATスレーブがあると、**不明なデバイス**として検出されます。
この場合、System Managerはデバイス内に保存されているESIを読み込むように要求します。これについては、チャプタ「[ESIデバイス記述ファイルに関する注記](#)」で説明します。
- ・ **デバイスが正常に検出されない**
考えられる理由：
 - データリンクの不備により、スキャン中にデータ損失が発生している
 - スレーブのデバイス記述ファイルが無効である
 接続やデバイスを緊急スキャンなどの適切な方法でチェックする必要があります。
その後、スキャンを再実行します。

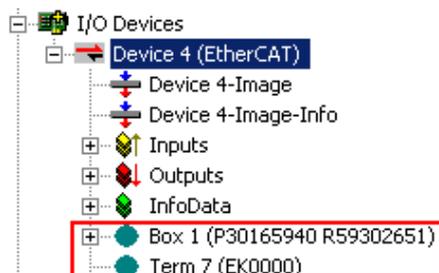


図 100: 識別の失敗

System Managerでは、これらのデバイスはEK0000または不明なデバイスとしてセットアップされることがあります。この場合、操作ができないか、操作が無効になります。

既存のコンフィグレーションに対するスキャン

注記

比較後のコンフィグレーションの変更

現状 (TwinCAT 2.11または3.1) では、スキャンすることによって比較されるデバイスプロパティはベンダ (メーカー)、デバイス名、およびリビジョンのみです。[ChangeTo]や[Copy]は、前述したベッコフのIO互換性ルールを認識した上で、慎重に実行する必要があります。比較後、デバイスコンフィグレーションが、検出されたリビジョンのコンフィグレーションで置換されます。これにより、サポートしているプロセスデータや機能に影響を受ける可能性があります。

既存のコンフィグレーションに対してスキャンを開始すると、実際のI/O環境がコンフィグレーションと正確に一致しているか、異なっているかが分かります。これにより、コンフィグレーションを比較できます。



図 101: 同一のコンフィグレーション (左: TwinCAT 2、右: TwinCAT 3)

検出された違いは修正ダイアログに表示され、ユーザは必要に応じてコンフィグレーションを修正することが可能です。

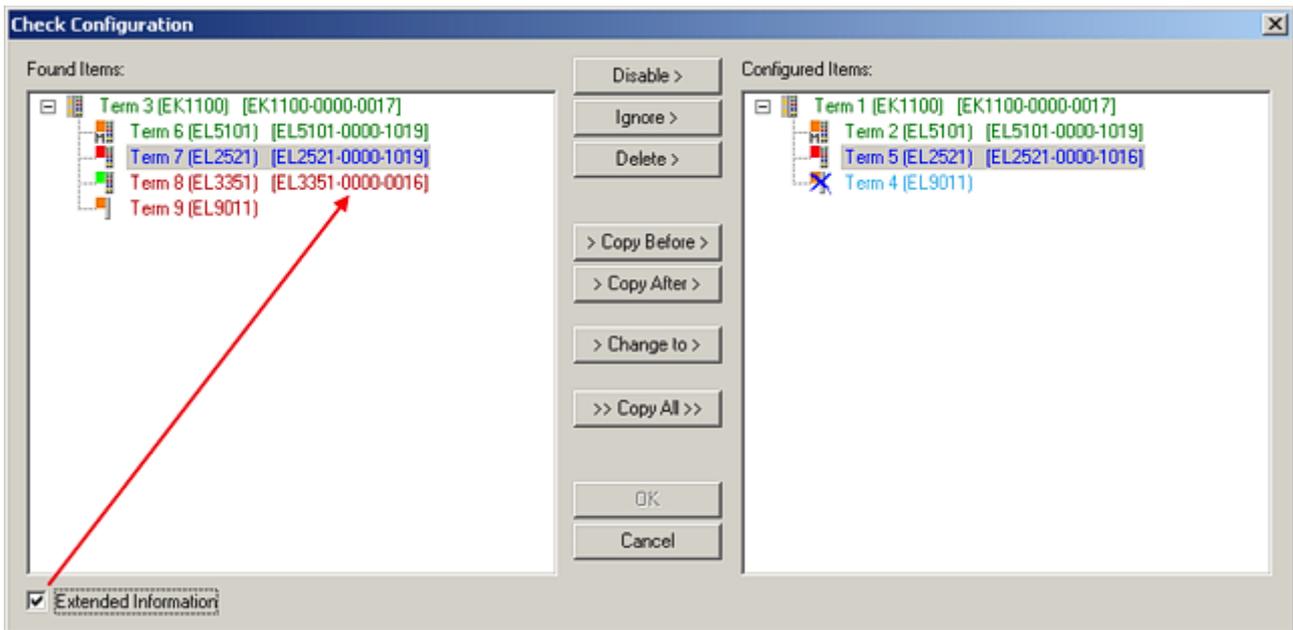


図 102: 修正ダイアログ

リビジョンの違いを確認する場合は、[Extended Information]チェックボックスにチェックを入れることを推奨します。

色	説明
緑	このEtherCATスレーブは、他方のエントリと一致しています。双方のタイプとリビジョンが一致しています。
青	このEtherCATスレーブは他方にも存在していますが、リビジョンが異なります。他方のリビジョンは、プロセスデータのデフォルト値、および機能が異なっている可能性があります。 設定されているリビジョンよりも検出されたリビジョンが上位である場合、互換性の問題を認識した上でスレーブを使用することができます。 設定されているリビジョンよりも検出されたリビジョンが下位である場合、スレーブを使用できない可能性があります。検出されたデバイスは、デバイスマスタが上位のリビジョン番号に基づいて使用できると想定しているすべての機能をサポートしていない可能性があります。
水色	このEtherCATスレーブは無視されます ([Ignore] ボタン)。
赤	<ul style="list-style-type: none"> このEtherCATスレーブは存在していません。 存在しているものの、リビジョンが異なっており、属性も指定されているものとは異なります。 ここでも、互換性のルールが適用されます。設定されているリビジョンよりも検出されたリビジョンが上位である場合、後継のデバイスが以前のデバイスの機能をサポートしているため、互換性の問題を認識した上で使用可能です。 設定されているリビジョンよりも検出されたリビジョンが下位である場合、スレーブを使用できない可能性があります。検出されたデバイスは、デバイスマスタが上位のリビジョン番号に基づいて使用できると想定しているすべての機能をサポートしていない可能性があります。

i リビジョンに基づいたデバイス選択、互換性

ESIファイルは、プロセスイメージ、マスタとスレーブ/デバイス間の通信タイプ、および適用可能な場合はデバイス機能も定義します。物理デバイス(使用可能な場合はファームウェア)は、マスタの通信確認ダイアログ/設定をサポートする必要があります。これは後方互換であり、EtherCATマスタが新しいデバイス(上位リビジョン)を古いリビジョンとして扱う場合でも、そのデバイスがサポートされます。ベッコフEtherCATターミナル/ボックス/EJモジュールは、以下の互換性を前提としています。

システム内のデバイスリビジョン ≧ コンフィグレーション内のデバイスリビジョン

これにより、コンフィグレーションを変更せずに、後でデバイスを交換することが可能です(ドライブに対して異なる指定が可能)。

例:

コンフィグレーションでEL2521-0025-1018が指定されている場合、実際にはEL2521-0025-1018以降(-1019、-1020)を使用できます。

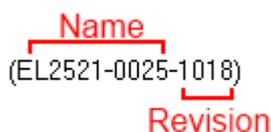


図 103: ターミナルの名前/リビジョン

TwinCATシステム内で最新のESIファイルが使用できる場合は、選択ダイアログで候補として表示される最新のリビジョンが、ベッコフの現行製品に適合します。実際のアプリケーションで最新のベッコフデバイスを使用する場合は、コンフィグレーションを新規作成する際に最新のデバイスリビジョンを使用することを推奨します。古いリビジョンは、在庫していた古いデバイスをアプリケーション内で使用する場合のみ使用してください。

この場合、デバイスのプロセスイメージがコンフィグレーションツリーに表示され、以下のパラメータ設定が可能になります：タスクとのリンク、CoE/DC設定、プラグイン定義、スタートアップ設定など。

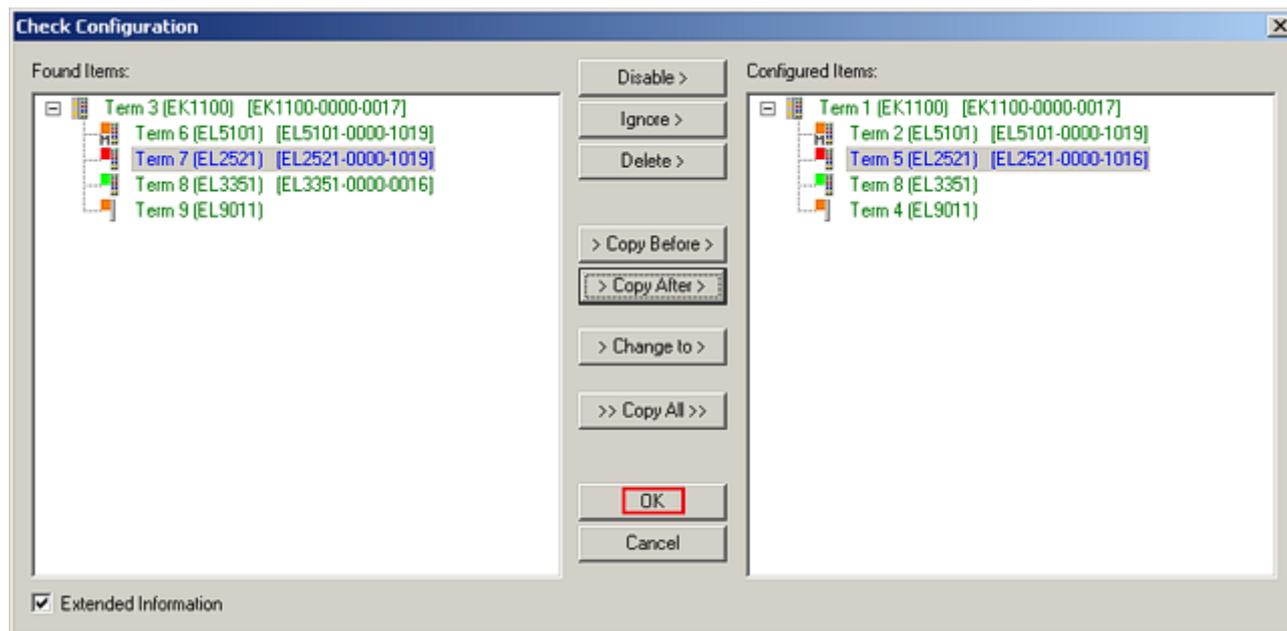


図 104: 修正ダイアログでの変更

すべての変更を保存または承認したら、[OK]をクリックして実際の*.tsmコンフィグレーションに転送します。

Change to Compatible Type

TwinCATには、タスク内のリンクを維持しつつデバイスを交換できるように、[Change to Compatible Type...]機能が用意されています。

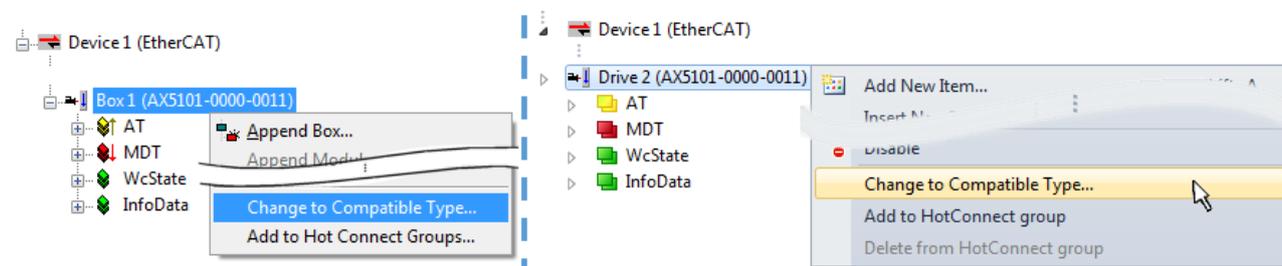


図 105: ダイアログ[Change to Compatible Type...] (左: TwinCAT 2、右: TwinCAT 3)

この機能は、AX5000デバイスでの使用を推奨します。

Change to Alternative Type

TwinCATには、デバイス交換のための[Change to Alternative Type]機能が用意されています。

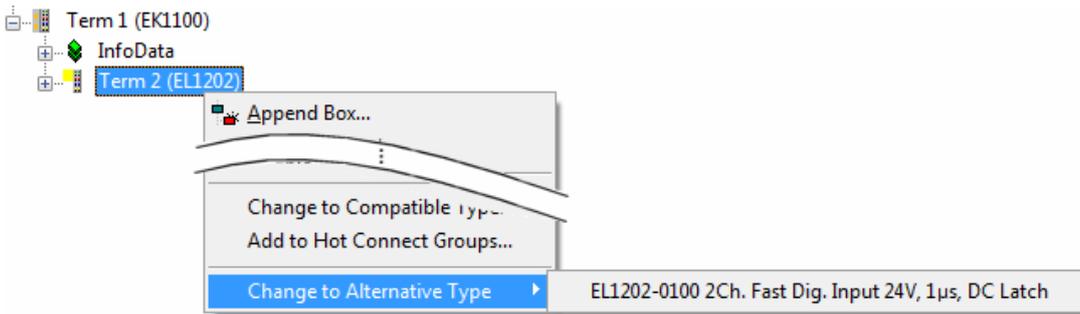


図 106: TwinCAT 2のダイアログ[Dialog Change to Alternative Type]

この機能を開始すると、System Managerは生成されたデバイスESI（この例ではEL1202-0000）に含まれている、互換性のあるデバイスの詳細を検索します。コンフィグレーションの変更と同時に、ESI-EEPROMが上書きされます。このため、この処理はオンライン状態(ConfigMode)でしか行えません。

5.2.5 EtherCATサブスクリバコンフィグレーション

TwinCAT 2 System Managerの左側のウィンドウ、またはTwinCAT 3開発環境のSolution Explorer内で、設定を行うツリー内のターミナルの元素(この例ではEL3751 Terminal 3)をクリックします。

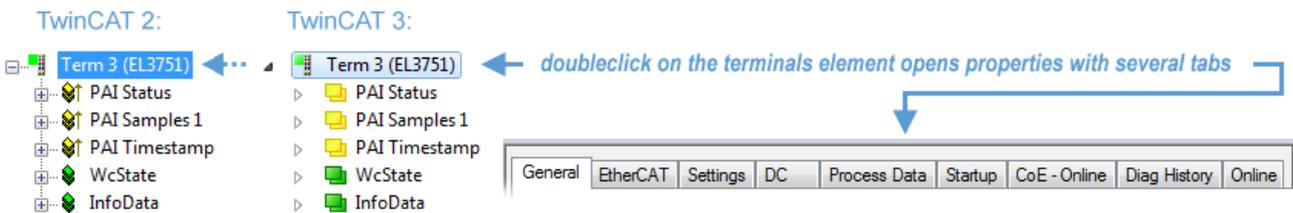


図 107: ターミナルEL3751としての分岐元素ト

TwinCAT System managerの右側のウィンドウ(TwinCAT 2)、または開発環境(TwinCAT 3)内には、ターミナルを設定するための各種タブが表示されます。表示されるタブは、サブスクリバの複雑さによって異なります。上図のように、ターミナルEL3751の場合は、多数のセットアップオプションと相当数のタブが表示されます。一方、例えばターミナルEL1004の場合は、[General]、[EtherCAT]、[Process Data]、[Online]のタブしか表示されません。EL6695など、特殊機能がターミナル名のタブ(EL6695の場合は[EL6695]タブ)で提供されるターミナルもあります。セットアップオプションが多岐にわたるターミナルの場合は、特殊タブ[Settings]が用意されていることもあります(EL3751など)。

[General]タブ



図 108: [General]タブ

Name	EtherCATデバイスの名前
Id	EtherCATデバイスの数
タイプ	EtherCATデバイスタイプ

Comment

ここで、コメントを追加できます(システムに関する情報など)。

Disabled

ここで、EtherCATデバイスを無効にできます。

Create symbols

このチェックボックスが有効な場合のみ、このEtherCATスレーブへのADS経由でのアクセスが可能です。

[EtherCAT]タブ

図 109: [EtherCAT]タブ

タイプ

EtherCATデバイスタイプ

Product/Revision

EtherCATデバイスの製品番号およびリビジョン番号

Auto Inc Addr.

EtherCATデバイスのオートインクリメントアドレス。オートインクリメントアドレスを使用して、通信リング内の物理的位置により各EtherCATデバイスをアドレス指定できます。オートインクリメントアドレス指定は、EtherCATマスタがEtherCATデバイスにアドレスを割り当てる際のスタートアップ段階で使用されます。オートインクリメントアドレス指定を使用すると、リング内の最初のEtherCATスレーブのアドレスは0000_{hex}となります。それ以降のスレーブに対するアドレスは、1ずつデクリメントされます(FFF_{hex}、FFF_{hex}など)。

EtherCAT Addr.

EtherCATスレーブの固定アドレス。このアドレスは、スタートアップ段階でEtherCATマスタによって割り当てられます。デフォルト値を変更するには、入力フィールドの左にあるチェックボックスにチェックを入れます。

Previous Port

このデバイスを接続しているEtherCATデバイスの名前とポート。通信リング内のEtherCATデバイスの順序を変更せずに、このデバイスを他のデバイスと接続することが可能な場合、この複合フィールドが有効になり、このデバイスに接続するEtherCATデバイスを選択できます。

Advanced Settings

このボタンをクリックすると、詳細設定のダイアログが開きます。

タブの下部には、このEtherCATデバイスのインターネット上の製品ページへのリンクが表示されます。

[Process Data]タブ

プロセスデータのコンフィグレーションを表示します。EtherCATスレーブの入出力データが、CANopenプロセスデータオブジェクト(Process Data Object、PDO)として表示されます。EtherCATスレーブがこの機能をサポートしている場合、ユーザはPDO Assignmentを使用してPDOを選択し、このダイアログで個々のPDOの内容を変更できます。

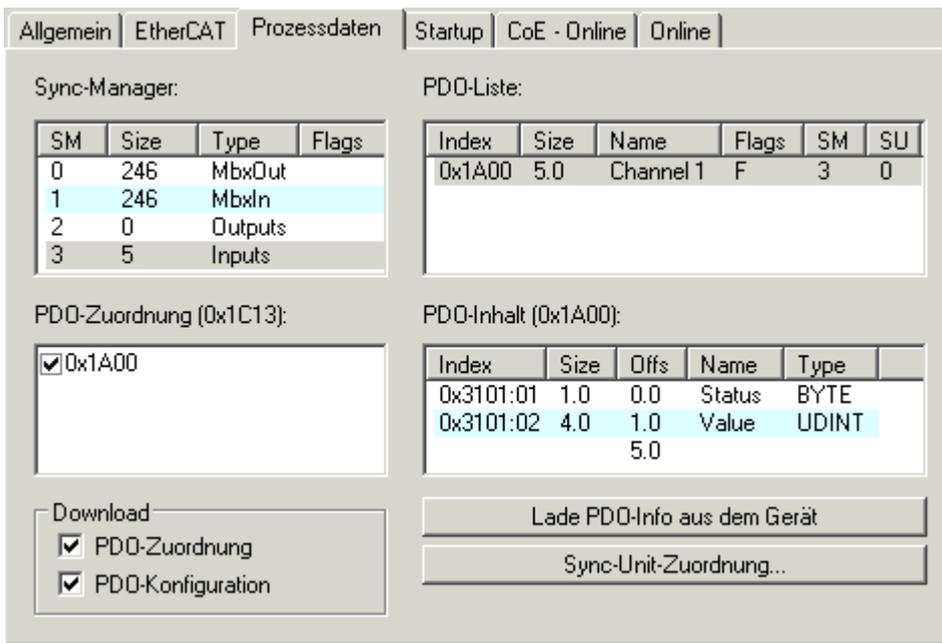


図 110: [Process Data] タブ

各サイクル中にEtherCATスレーブによって転送されるプロセスデータ (PDO) はユーザデータであり、アプリケーションはこのデータが周回的に更新され、スレーブに送信されることを想定しています。これを行うために、EtherCATマスタ (ベッコフTwinCAT) はスタートアップ段階で各EtherCATスレーブをパラメータ設定し、EtherCATマスタがこのスレーブに、またはこのスレーブから転送するプロセスデータ (ビット/バイト単位のサイズ、データの場所、送信タイプ) を定義します。設定が誤っていると、スレーブが正常にスタートアップしない可能性があります。

ベッコフEtherCAT EL、ES、EM、EJおよびEPスレーブには、原則的に以下が適用されます。

- ・ デバイスによってサポートする入出力プロセスデータは、メーカーによってESI/XMLファイルの形式で定義されます。TwinCAT EtherCATマスタはESIファイルを使用して、スレーブを正しく設定します。
- ・ プロセスデータはSystem Manager内で変更できます。デバイスの取扱説明書を参照してください。
変更例: チャンネルのマスク、追加の周期情報の表示、8ビットではなく16ビットデータサイズでの表示など
- ・ いわゆる高機能なEtherCATデバイスでは、プロセスデータ情報もCoEディレクトリに保存されます。CoEディレクトリ内のPDO設定が変わってしまうようないかなる変更は、正常なスレーブのスタートアップが妨げられます。デバイスのファームウェア (利用可能な場合) はこれらのPDOの組み合わせで動作するようになっているため、指定されたものと異なるプロセスデータを使用することは推奨されません。

デバイスの取扱説明書でプロセスデータの変更が許可されている場合は、以下の手順にしたがってください (図. 「プロセスデータの設定」を参照)。

- ・ A: 設定するデバイスを選択します。
- ・ B: [Process Data] タブの [SyncManager] (C) で、[Input] または [Output] を選択します。
- ・ D: PDOを選択、または選択解除できます。
- ・ H: System Manager内に、リンク可能な変数として新しいプロセスデータが表示されます。新しいプロセスデータは、コンフィグレーションが有効になり、TwinCATを再起動する (またはEtherCATマスタを再起動する) とアクティブになります。
- ・ E: スレーブがこれをサポートしている場合、いわゆるPDOレコード ([Predefined PDO settings]) を選択することで、入力および出力PDOを同時に変更できます。

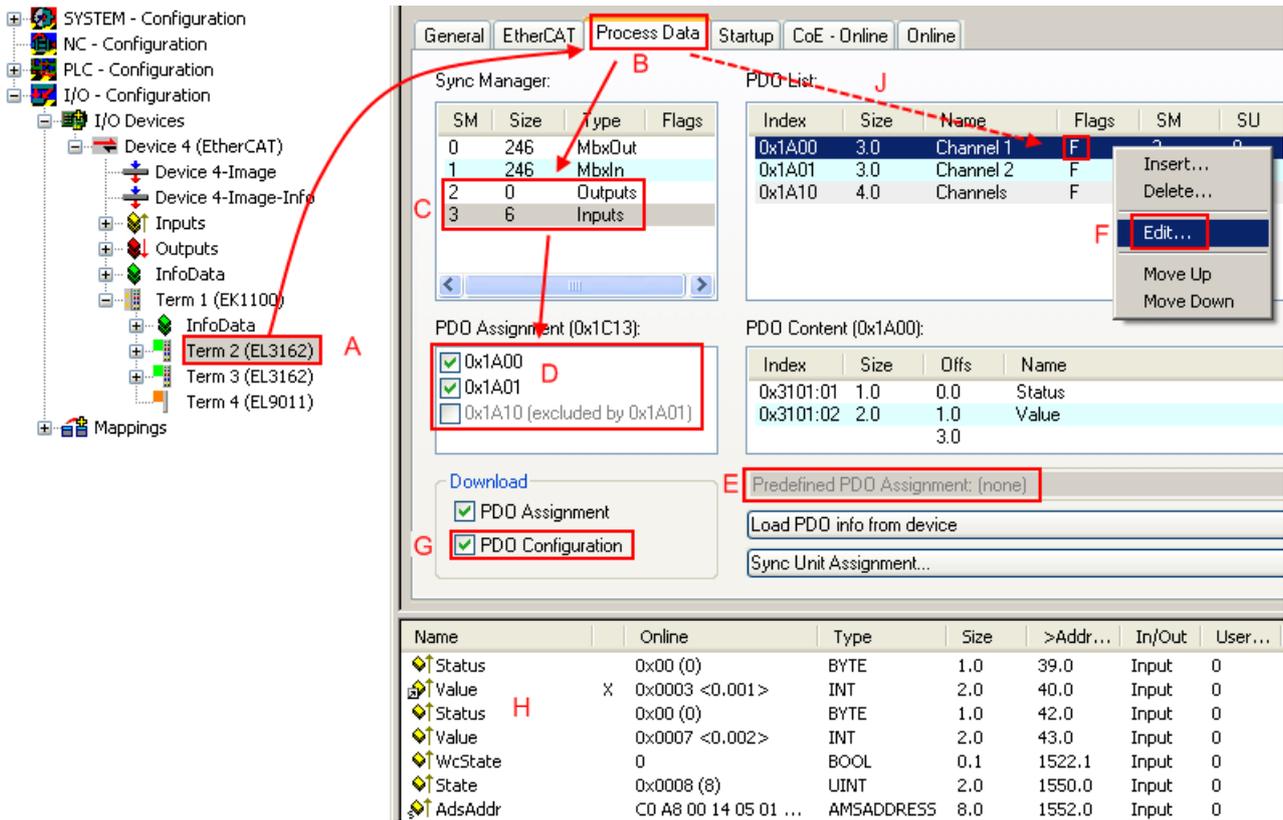


図 111: プロセスデータの設定

● プロセスデータの手動での変更

i ESIファイルに基づき、PDO一覧内でフラグ「F」が付けられているPDOは「固定」のPDOとして識別されます(図、「プロセスデータの設定」、J)。このようなPDOのコンフィギュレーションは、関連するダイアログ([Edit])をTwinCATが表示した場合でも変更できません。特に、CoEの内容は周期プロセスデータとして表示できません。通常、これはデバイスがPDOコンフィギュレーション(「G」)のダウンロードをサポートしている場合にも当てはまります。コンフィギュレーションに誤りがある場合、通常、EtherCATスレーブは開始およびOP状態への変更を拒否します。System Managerは「invalid SM cfg」というログメッセージを表示します。このエラーメッセージ(「invalid SM IN cfg」または「invalid SM OUT cfg」)は、スタートアップが失敗した理由を表しています。

詳細な説明 [▶ 86]は、このセクションの最後に記載されています。

[Startup]タブ

EtherCATスレーブにメールボックスがあり、CANopen over EtherCAT (CoE)またはServo drive over EtherCATプロトコルをサポートしている場合は、[Startup]タブが表示されます。このタブには、スタートアップ中にメールボックスに送信されるダウンロードリクエストが表示されます。リスト表示に新しいメールボックスリクエストを追加することも可能です。ダウンロードリクエストは、リストに表示されている順序でスレーブに送信されます。

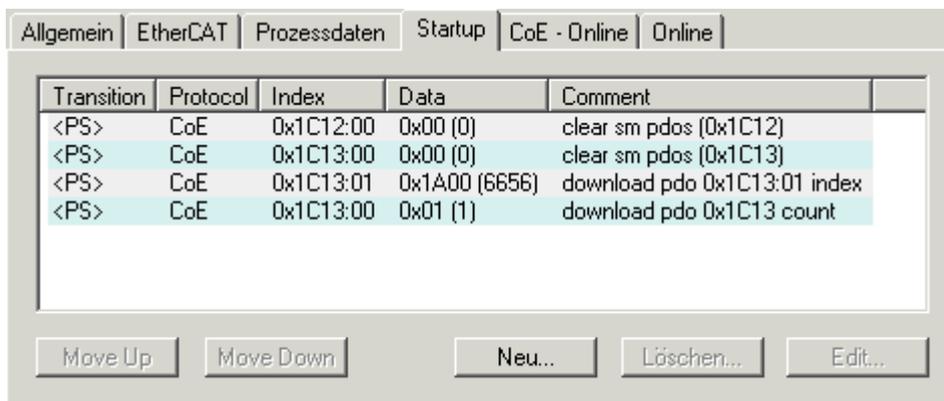


図 112: [Startup]タブ

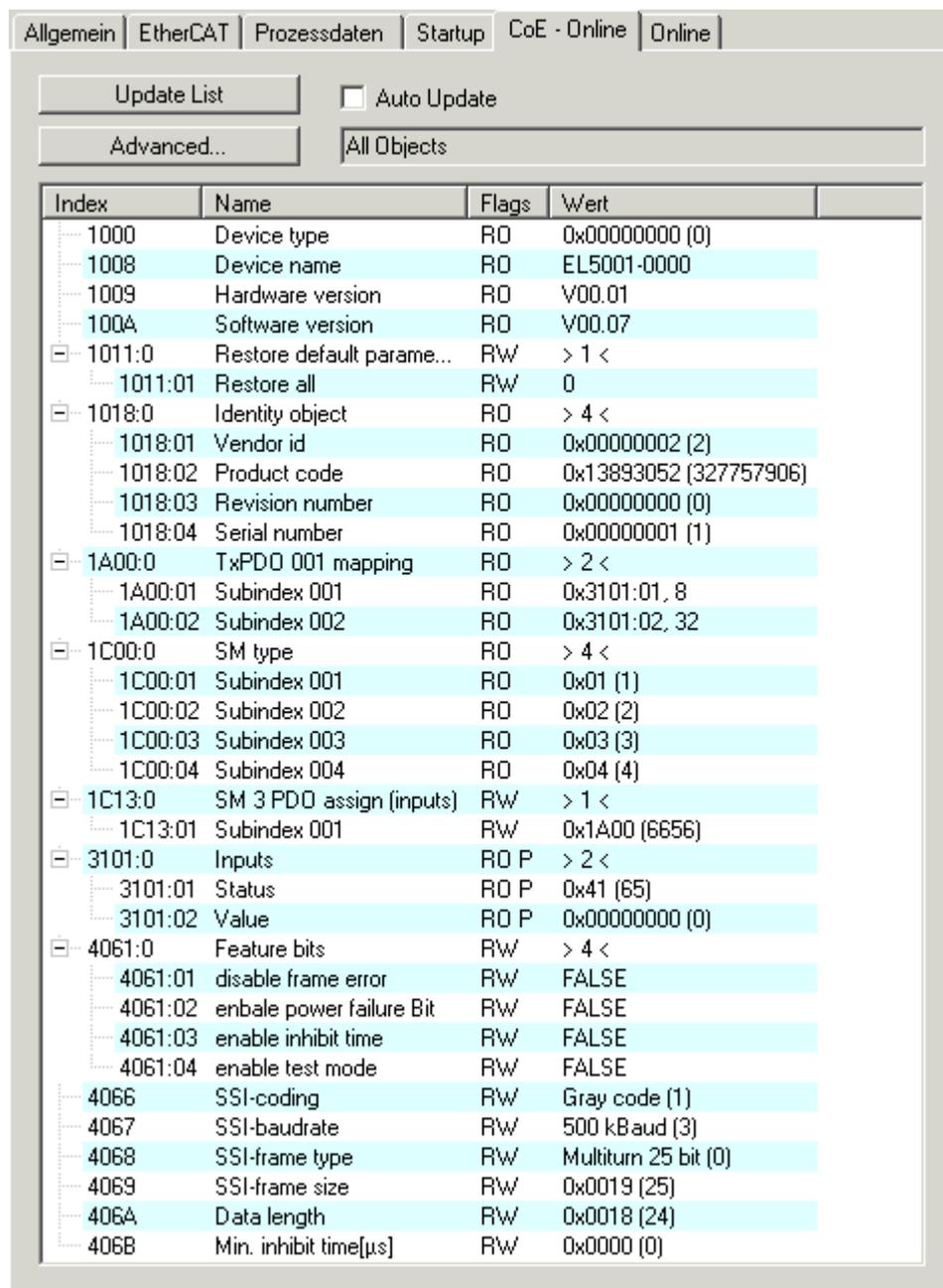


図 113: [CoE - Online] タブ

オブジェクトリスト表示

列	説明	
Index	オブジェクトのインデックスおよびサブインデックス	
Name	オブジェクト名	
Flags	RW	オブジェクトを読み取ることができ、かつデータをオブジェクトに書き込むことができます (読み取り/書き込み)。
	RO	オブジェクトを読み取ることにはできますが、データをオブジェクトに書き込めません (読み取り専用)。
	P	Pが追加されている場合は、オブジェクトがプロセスデータオブジェクトであることを意味します。
Value	オブジェクトの値	

- Update List** [Update list] ボタンは、表示されているリスト内のオブジェクトをすべて更新します。
- Auto Update** このチェックボックスを選択すると、オブジェクトの内容が自動的に更新されます。
- Advanced** [Advanced] ボタンにより、[Advanced Settings] ダイアログが開きます。このダイアログで、表示するオブジェクトのリストを指定できます。

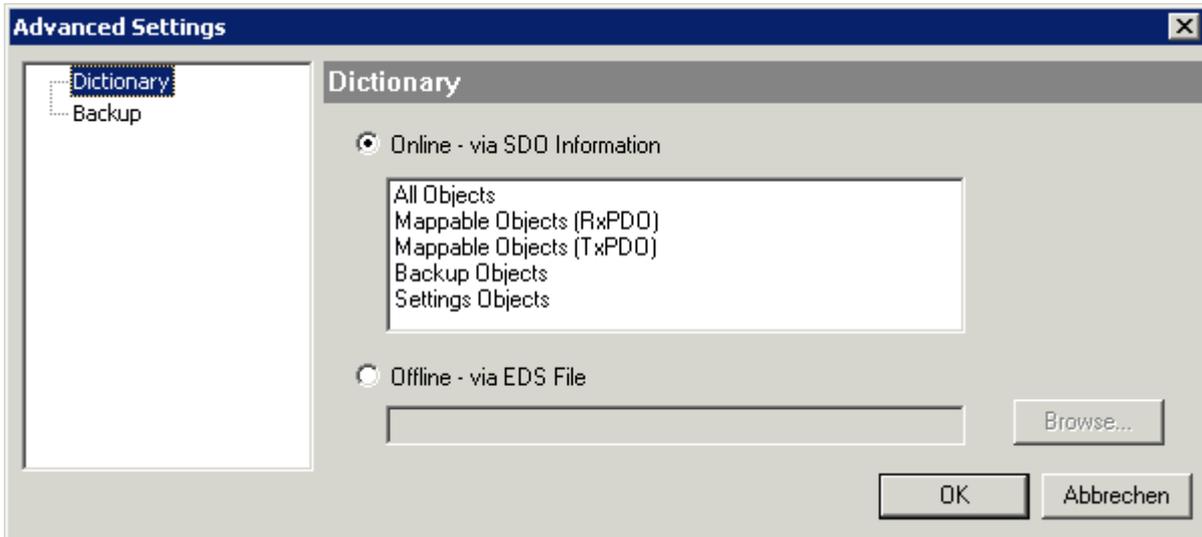


図 114: [Advanced settings] ダイアログ

- Online - via SDO Information** このオプションボタンが選択されていると、スレーブのオブジェクトリストに含まれているオブジェクトのリストが、SDO Information 経由でスレーブからアップロードされます。下部のリストを使用して、どのオブジェクトタイプをアップロードするかを指定できます。
- Offline - via EDS File** このオプションボタンが選択されていると、オブジェクトリストに含まれているオブジェクトのリストが、ユーザが用意した EDS (ESI) ファイルから読み込まれます。

[Online] タブ

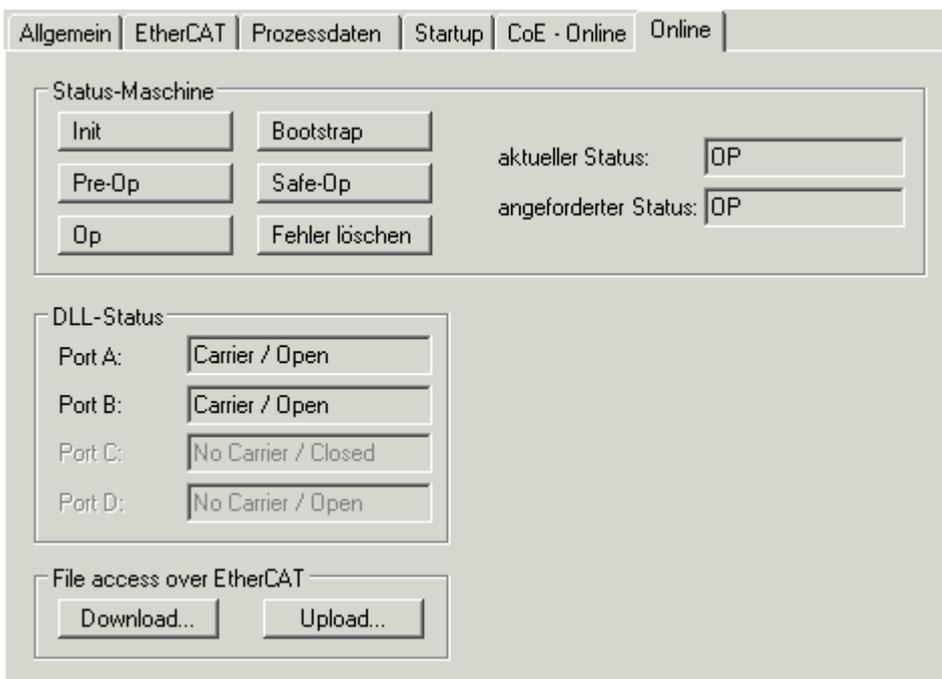


図 115: [Online] タブ

ステートマシン

Init	このボタンを押すと、EtherCATデバイスの <i>Init</i> 状態へのセットを試行します。
Pre-Op	このボタンを押すと、EtherCATデバイスの <i>Pre-operational</i> 状態へのセットを試行します。
Op	このボタンを押すと、EtherCATデバイスの <i>Operational</i> 状態へのセットを試行します。
Bootstrap	このボタンを押すと、EtherCATデバイスの <i>Bootstrap</i> 状態へのセットを試行します。
Safe-Op	このボタンを押すと、EtherCATデバイスの <i>Safe-operational</i> 状態へのセットを試行します。
Clear Error	このボタンを押すと、エラー表示の消去を試行します。状態の変更中にEtherCATスレーブに障害が発生すると、このスレーブはエラーフラグをセットします。 例：EtherCATスレーブがPREOP状態 (Pre-operational) にあります。ここで、マスタがSAFEOP状態 (Safe-operational) をリクエストします。状態の変更中にこのスレーブに障害が発生すると、スレーブはエラーフラグをセットします。現在の状態が、ERR PREOPとして表示されます。[Clear Error] を押すと、エラーフラグがクリアされ、現在の状態が再度PREOPとして表示されます。
Current State	EtherCATデバイスの現在の状態を表示します。
Requested State	EtherCATデバイスに対してリクエストされた状態を表示します。

DLL Status

EtherCATスレーブの個々のポートのDLLステータス（データリンク層のステータス）を表示します。DLLステータスには、以下の4つの状態があります。

ステータス	説明
No Carrier / Open	ポートで「キャリアなし」信号が検出されませんが、ポートは開いています。
No Carrier / Closed	ポートで「キャリアなし」信号が検出され、ポートが閉じています。
Carrier / Open	ポートで「キャリア」信号が検出され、ポートが開いています。
Carrier / Closed	ポートで「キャリア」信号が検出されませんが、ポートが閉じています。

File Access over EtherCAT

Download	このボタンを押すと、ファイルをEtherCATデバイスに書き込みます。
Upload	このボタンを押すと、ファイルをEtherCATデバイスから読み込みます。

[DC] タブ (ディストリビュートクロック)

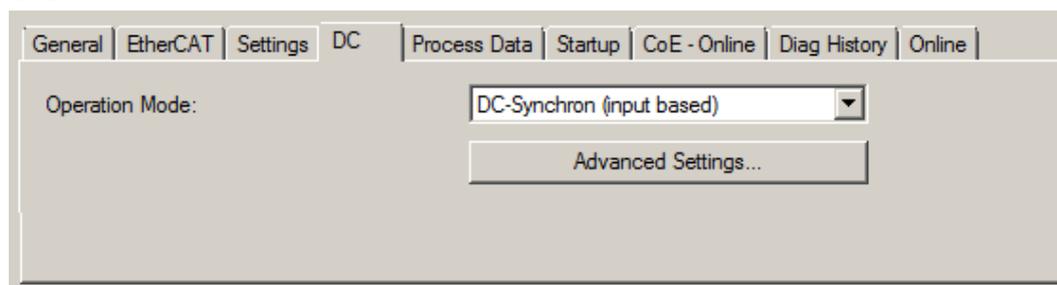


図 116: [DC] タブ (ディストリビュートクロック)

Operation Mode	オプション (オプション): <ul style="list-style-type: none"> ・ FreeRun ・ SM-Synchron ・ DC-Synchron (Input based) ・ DC-Synchron
Advanced Settings...	リアルタイムかつ時間確定性のあるTwinCATクロックを再調整するための詳細設定

ディストリビュートクロックの詳細情報は、<http://infosys.beckhoff.com>を参照してください。

Fieldbus Components → EtherCAT Terminals → EtherCAT System documentation → EtherCAT basics → Distributed Clocks

5.2.5.1 [Process Data] タブの詳細

Sync Manager

Sync Manager (SM) のコンフィグレーションを表示します。
 EtherCAT デバイスにメールボックスがある場合は、メールボックス出力 (MbxOut) に SM0、メールボックス入力 (MbxIn) に SM1 が使用されます。
 出力プロセスデータには SM2 (出力) が使用され、入力プロセスデータには SM3 (入力) が使用されます。
 入力が選択されている場合、対応する PDO 割り当てが下部の [PDO Assignment] リストに表示されます。

PDO Assignment

選択した Sync Manager の PDO 割り当て。この Sync Manager に対して定義されているすべての PDO がここに表示されます。

- ・ [Sync Manager] リスト内で出力 Sync Manager (出力) が選択されている場合は、すべての RxPDO が表示されます。
- ・ [Sync Manager] リスト内で入力 Sync Manager (入力) が選択されている場合は、すべての TxPDO が表示されます。

選択したエントリが、プロセスデータ通信に含まれる PDO となります。System Manager のツリーに、これらの PDO が EtherCAT デバイスの変数として表示されます。変数名は、PDO リストに表示される PDO の Name パラメータと同一です。PDO 割り当てリスト内のエントリが無効 (選択されておらずグレースアウト) の場合は、この入力が PDO 割り当てから除外されていることを意味します。グレースアウトされている PDO を選択するには、先に現在選択されている PDO の選択を解除する必要があります。



PDO 割り当ての有効化

- ✓ PDO 割り当てを変更した場合、新しい PDO 割り当てを有効にするには、以下を実行する必要があります。
 - a) 一度、EtherCAT スレーブで PS ステータス遷移サイクル (Pre-operational から Safe-operational) を実行する必要があります (「Online tab」▶ 84) を参照)。
 - b) さらに、System Manager で EtherCAT スレーブをリロードする必要があります。

(TwinCAT 2 の  ボタン、または TwinCAT 3 の  ボタン)

PDO list

この EtherCAT デバイスがサポートしているすべての PDO のリスト。選択されている PDO の内容が、[PDO Content] リストに表示されます。任意のエントリをダブルクリックすることで、PDO コンフィグレーションを変更できます。

列	説明	
Index	PDO のインデックス。	
Size	バイト単位の PDO のサイズ。	
Name	PDO の名前。 この PDO が Sync Manager に割り当てられている場合、このパラメータと同名のスレーブの変数として表示されます。	
フラグ	F	固定された内容: この PDO の内容は固定されており、System Manager で変更できません。
	M	必須 PDO。この PDO は必須であるため、Sync Manager に割り当てる必要があります。このため、この PDO は [PDO Assignment] リストから削除できません。
SM	この PDO が割り当てられる Sync Manager。このエントリが空の場合、この PDO はプロセスデータ通信に関与しません。	
SU	この PDO が割り当てられているシンクユニット。	

PDO Content

PDO の内容を表示します。PDO にフラグ F (固定された内容) がセットされていない場合、内容を変更できません。

Download

デバイスが「インテリジェント」であり、かつメールボックスがある場合は、PDO のコンフィグレーションおよび PDO 割り当てをデバイスにダウンロードできます。これはオプション機能であり、この機能をサポートしていない EtherCAT スレーブもあります。

PDO Assignment

このチェックボックスが選択されている場合、[PDO Assignment]リスト内で設定したPDO割り当てがスタートアップ時にデバイスにダウンロードされます。デバイスに送信する必要なコマンドは、[Startup [▶ 81]]タブに確認できます。

PDO Configuration

このチェックボックスが選択されている場合、対応するPDOのコンフィグレーション (PDOリストおよび[PDO Content])に表示)がEtherCATスレーブにダウンロードされます。

5.3 一般的な注意 - EtherCATスレーブアプリケーション

ここでは、TwinCATでのEtherCATスレーブの動作について簡単に説明します。より詳細な情報は、「EtherCATシステムマニュアル」などの該当するセクションに記載されています。

リアルタイムでの診断: WorkingCounter、EtherCAT状態およびステータス

一般的に、EtherCATスレーブはタスクの制御に使用可能なさまざまな診断情報を提供します。

この診断情報は、通信の複数のレベルに関連しています。このため、診断の情報源は多岐にわたり、更新のタイミングもさまざまです。

フィールドバスからの正常かつ最新のI/Oデータに依存するアプリケーションは、対応する下層の通信層の診断情報にアクセスできるようにする必要があります。EtherCATおよびTwinCATシステムマネージャには、この種類の包括的な診断エレメントが用意されています。運用時(コミッショニング中ではない)に現在のサイクルが正確かどうかを診断する制御タスクに役立つこれらの診断エレメントについて、以下で説明します。

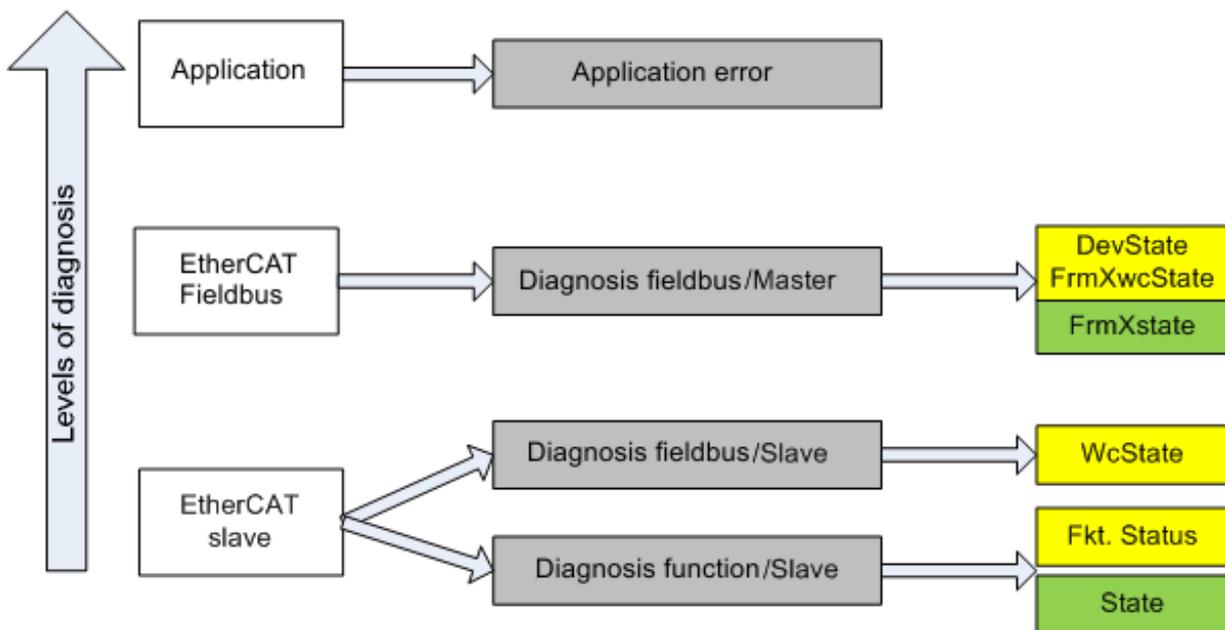


図 117: EtherCATスレーブの診断情報の選択

一般的に、EtherCATスレーブは以下を提供します。

- ・ スレーブに関する通信診断(プロセスデータの交換に正しく関与したか、および動作モードが正しいかを診断)
この診断は、すべてのスレーブで同一です。
- ・ チャンネルに関する機能診断(デバイス依存)
対応するデバイスの取扱説明書を参照してください。

図. 「EtherCATスレーブの診断情報の選択」の色は、System Managerでの色に対応しています。図. 「PLCによる基本的なEtherCATスレーブ診断」を参照してください。

色	意味
黄色	スレーブからEtherCATマスタへの入力変数、サイクル毎に更新

色	意味
赤	EtherCATマスタからスレーブへの出力変数、サイクル毎に更新
緑	非同期に更新されるEtherCATマスタ用の情報変数です。これは、特定のサイクル内で最新のステータスを表していない可能性があることを意味します。ADS経由でこれらの変数を読み込むと便利です。

図. 「PLCによる基本的なEtherCATスレーブ診断」は、基本的なEtherCATスレーブ診断の実行例を示しています。ここでは、スレーブに関する通信診断と、チャンネルに関する機能診断の両方を提供するベッコフEL3102 (2チャンネルアナログ入力ターミナル)を使用します。構造はPLC内で入力変数として作成されており、それぞれがプロセスイメージと対応しています。

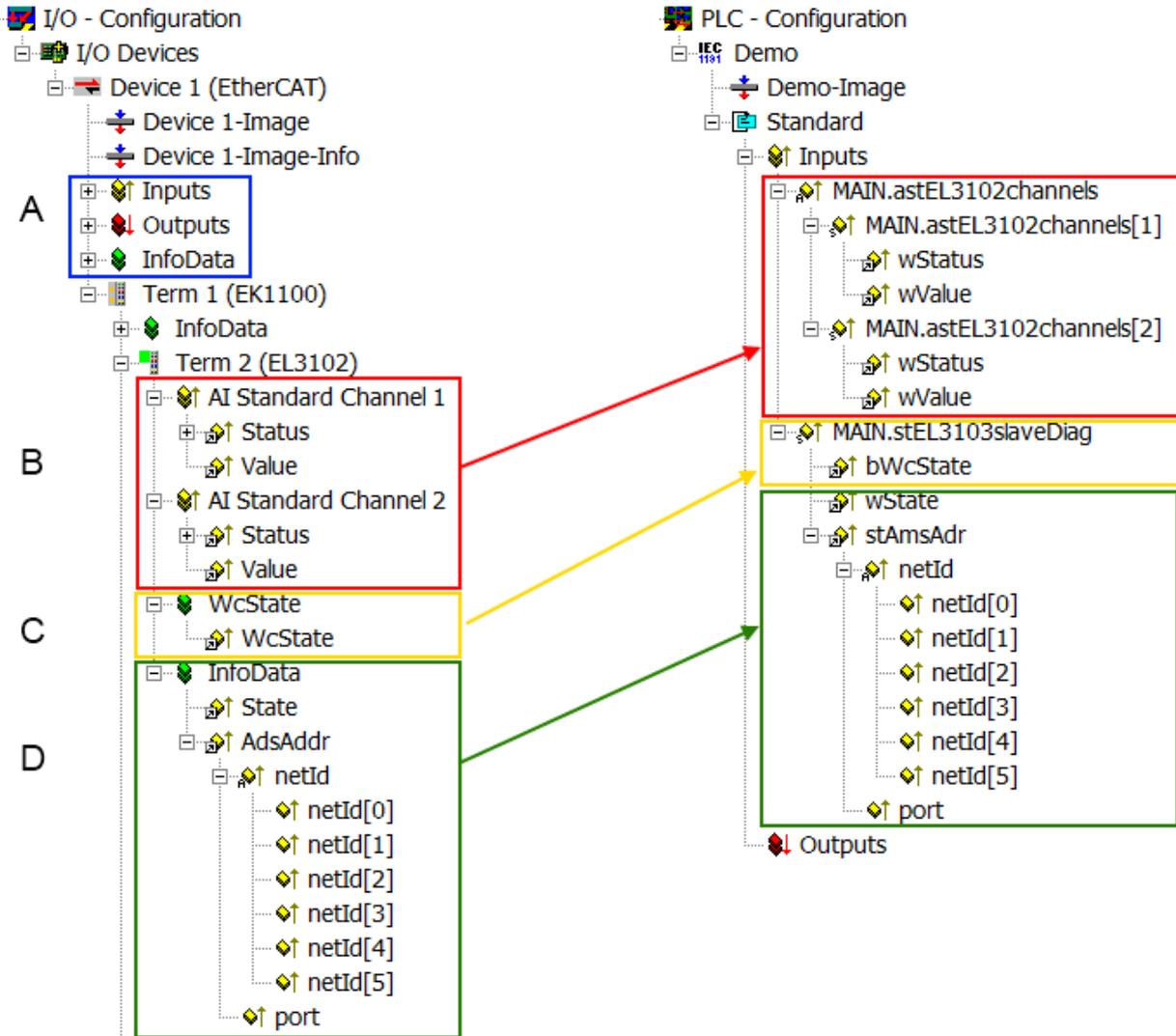


図 118: PLCによる基本的なEtherCATスレーブ診断

ここでは、以下について扱います。

コード	機能	実装	アプリケーション/評価
A	EtherCATマスタの診断情報 非周期的に更新(黄)、または非周期的に提供(緑)		PLC内の直近のサイクルについて、少なくともDevStateを評価します。 EtherCATマスタの診断情報には、「EtherCATシステムマニュアル」には記載されていない情報も含まれています。以下は、情報の例です。 <ul style="list-style-type: none"> スレーブを使用/経由する通信のためのマスタ内のCoE TcEtherCAT.libの機能 OnlineScanの実行

コード	機能	実装	アプリケーション/評価
B	この例 (EL3102) では、EL3102に対して、直近のサイクルに単一の機能ステータスを送信する2つのアナログ入力チャンネルが構成されています。	ステータス <ul style="list-style-type: none"> ビットの意味については、デバイスの取扱説明書に記載されている場合があります。 スレーブ以外のデバイスの場合、より多くの情報が提供されている場合がありますが、スレーブの場合、通常は情報がありません。 	上位レベルのPLCタスク (または対応する制御アプリケーション) が正常なデータを使用できるようにするには、そこで機能ステータスを評価する必要があります。このため、このような情報が最新のサイクルのプロセスデータとともに提供されます。
C	周期プロセスデータをもつ各EtherCATスレーブについて、マスタはいわゆる WorkingCounter を使用して、スレーブが周期的なプロセスデータ交換にエラーなく正常に関与しているかを表示します。このため、この重要かつ基本的な情報が System Manager 内で、直近のサイクルについて <ol style="list-style-type: none"> EtherCATスレーブにおいては同一内容で EtherCATマスタにおいては共通変数として (上図内のAを参照) リンク用に提供されます。	WcState (WorkingCounter) 0: 最後のサイクルでのリアルタイム通信が正常 1: リアルタイム通信が異常 これは、同一のシンクユニット内に配置されている他のスレーブのプロセスデータに影響を与える可能性があります。	上位レベルのPLCタスク (または対応する制御アプリケーション) が正常なデータを使用できるようにするには、そこでEtherCATスレーブの通信ステータスを評価する必要があります。このため、このような情報が最新のサイクルのプロセスデータとともに提供されます。
D	EtherCATマスタの診断情報は、スレーブのところにリンク状態としても表示され、そのスレーブの状態をマスタによって判定します。この情報は以下の理由でリアルタイム性がありません。 <ul style="list-style-type: none"> システムのスタートアップ時以外ほとんど変更されない、またはまったく変更されないため。 それ自身を非周期的に判定しているため (EtherCATステータスなど)。 	状態 スレーブの現在のステータス (INIT ~ OP)。正常に動作している場合、スレーブはOP (=8) である必要があります。 AdsAddr ADSアドレスは、CoEから、またはCoEへの読み取り/書き込みなど、EtherCATスレーブを使用したADS経由でのPLC/タスクからの通信に使用します。スレーブのAMSNetIDは、EtherCATマスタのAMSNetIDと対応します。個々のスレーブとの通信は、ポート (= EtherCATアドレス) で行うことができます。	非同期に更新されるEtherCATマスタ用の情報変数です。これは、特定のサイクル内で最新のステータスを表していない可能性があることを意味します。ADS経由でこれらの変数を読み込むことができます。

注記

診断情報
 アプリケーションが適切に反応できるように、利用可能な診断情報を評価することを強く推奨します。

CoEパラメータディレクトリ

CoE (CANopen over EtherCAT)パラメータディレクトリは、関連するスレーブの設定値の管理に使用されません。状況によっては、比較的设置が複雑なEtherCATスレーブはここで変更を行行わなければならないことがあります。TwinCAT System Managerからアクセス可能です。図. 「EL3102、CoEディレクトリ」を参照してください。

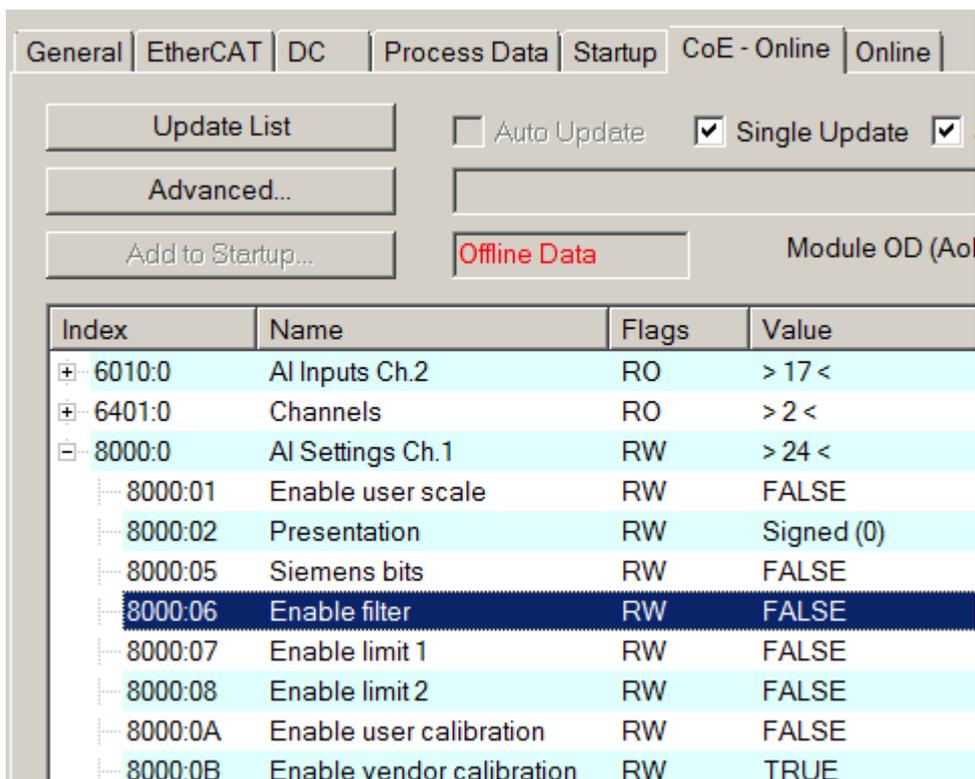


図 119: EL3102、CoEディレクトリ

● EtherCAT System Documentation

i 「EtherCAT System Documentation」 (EtherCAT Basics --> CoE Interface)に記載されている包括的な説明を順守する必要があります。

以下は抜粋です。

- ・ オンラインディレクトリ内の変更がスレーブ内にローカルに保存されるかどうかは、デバイスによって異なります。ELターミナル(EL66xxを除く)は、ローカルに保存することが可能です。
- ・ ユーザは、StartUpリストに対する変更を管理する必要があります。

TwinCAT System Managerのコミッショニング支援

コミッショニングインターフェイスは、EL/EP EtherCATデバイスの開発中のプロセスの一部として実装されています。これらのインターフェイスは、TwinCAT 2.11R2以降のTwinCAT System Managerで使用可能です。コミッショニングインターフェイスは、適切に拡張されたESIファイルによってSystem Managerに統合されます。

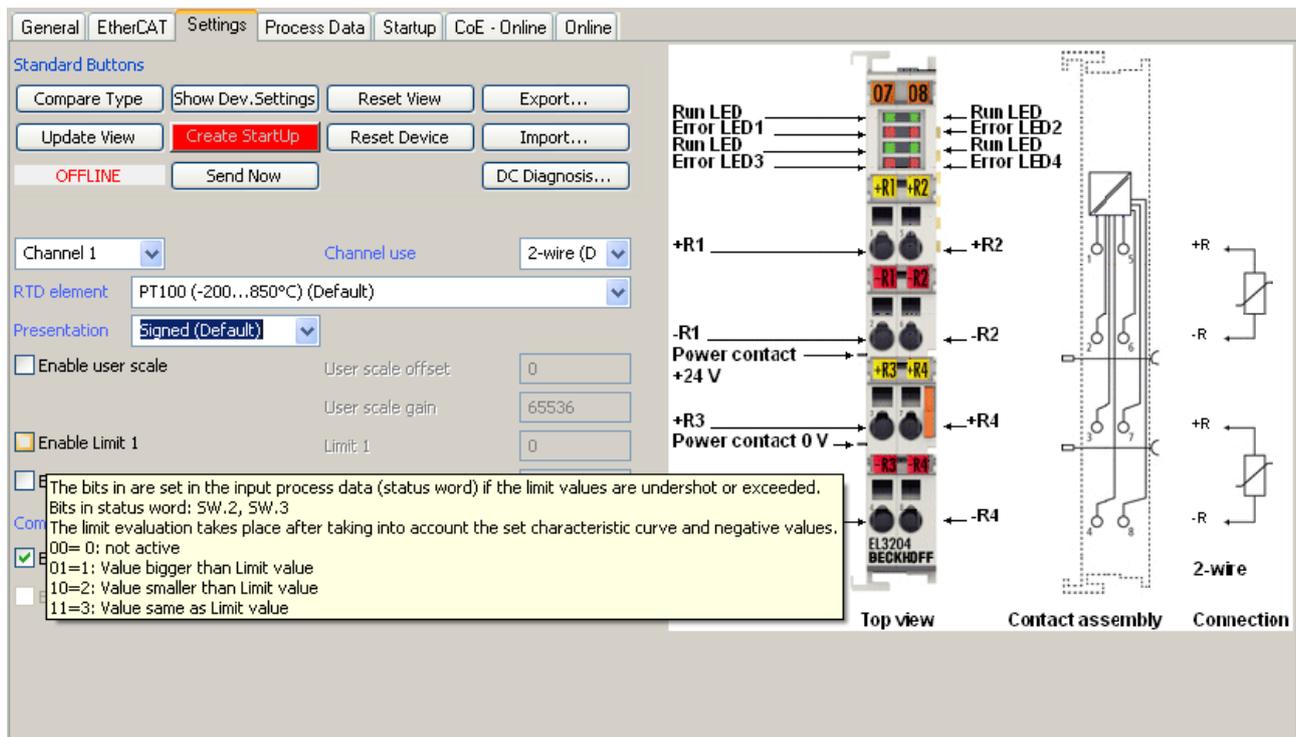


図 120: EL3204のコミッションング支援の例

このコミッションングプロセスは、以下を同時に処理します。

- ・ CoEパラメータディレクトリ
- ・ DC/FreeRunモード
- ・ 使用可能なプロセスデータレコード (PDO)

従来必要であった [Process Data]、[DC]、[Startup]、および [CoE-Online] も引き続き表示されますが、コミッションング支援を使用する場合は、自動生成された設定をコミッションング支援で変更しないことを推奨します。

コミッションングツールは、EL/EPデバイスで可能なすべての用途に対応している訳ではありません。使用可能な設定オプションでは足りない場合、ユーザは従来どおりDC、PDO、およびCoEを手動で設定できます。

EtherCAT状態: TwinCAT System Managerおよび手動操作の自動でのデフォルト動作

電源スイッチがオンになると、EtherCATスレーブは以下のステータスに移行します。

- ・ INIT
- ・ PREOP
- ・ SAFEOP
- ・ OP

EtherCATマスタは、デバイスのコミッションングに対してES/XMLおよびユーザ設定(ディストリビュートクロック (DC)、PDO、CoE)で定義された初期化ルーチンにしたがって、これらのステータスに対して指示を与えます。これに関しては、「[通信、EtherCATステートマシン](#)」の原則」のセクションも参照してください。実行の必要があるコンフィグレーションの量、および全体の通信量によっては、起動に数秒かかることがあります。

EtherCATマスタ自体は、OPターゲット状態に到達するまで、これらのルーチンを行う必要があります。

ユーザが望む、TwinCATによってスタートアップ時に自動的に移行するターゲット状態は、System Managerで設定できます。TwinCATがステータスRUNに到達すると、TwinCAT EtherCATマスタはすぐにターゲット状態に移行します。

標準設定

EtherCATマスタの詳細設定 (Advanced Settings) は、全てのスレーブに標準として設定されます。

- ・ EtherCATマスタ: OP
- ・ スレーブ: OP
この設定は、すべてのスレーブに等しく適用されます。

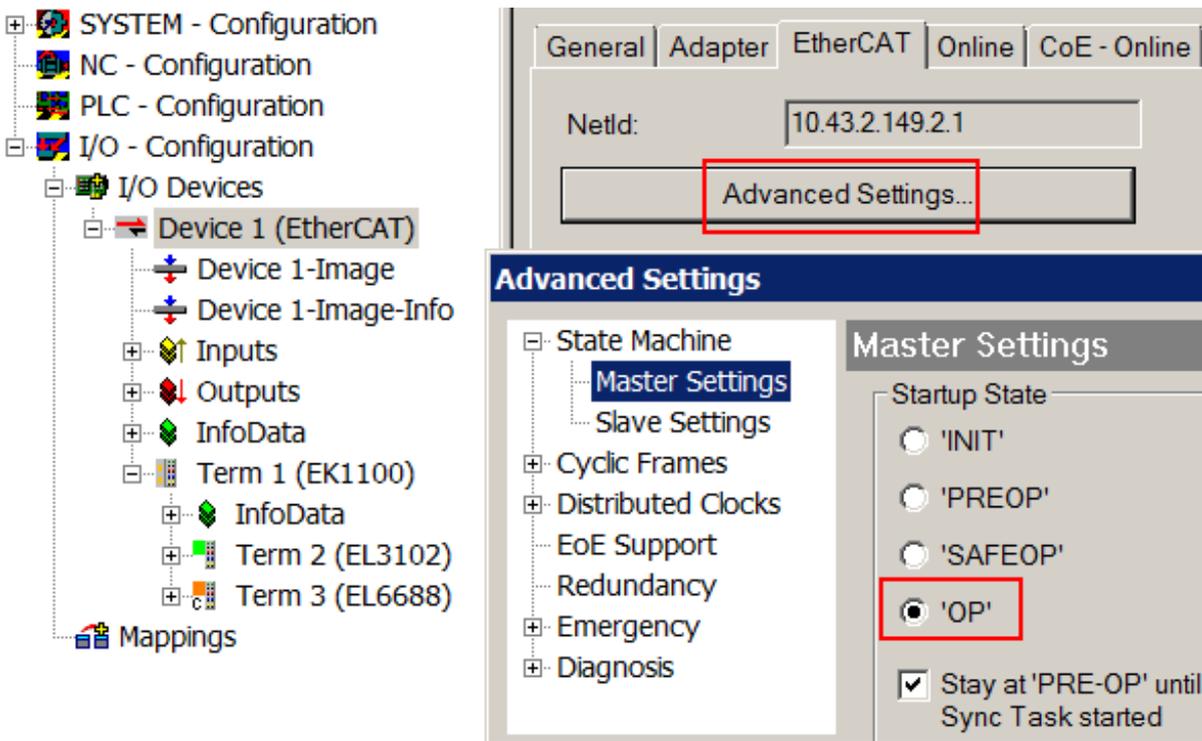


図 121: System Managerのデフォルト動作

加えて、任意のスレーブのターゲット状態を[Advanced Settings]ダイアログで設定できます。この場合も標準設定はOPです。

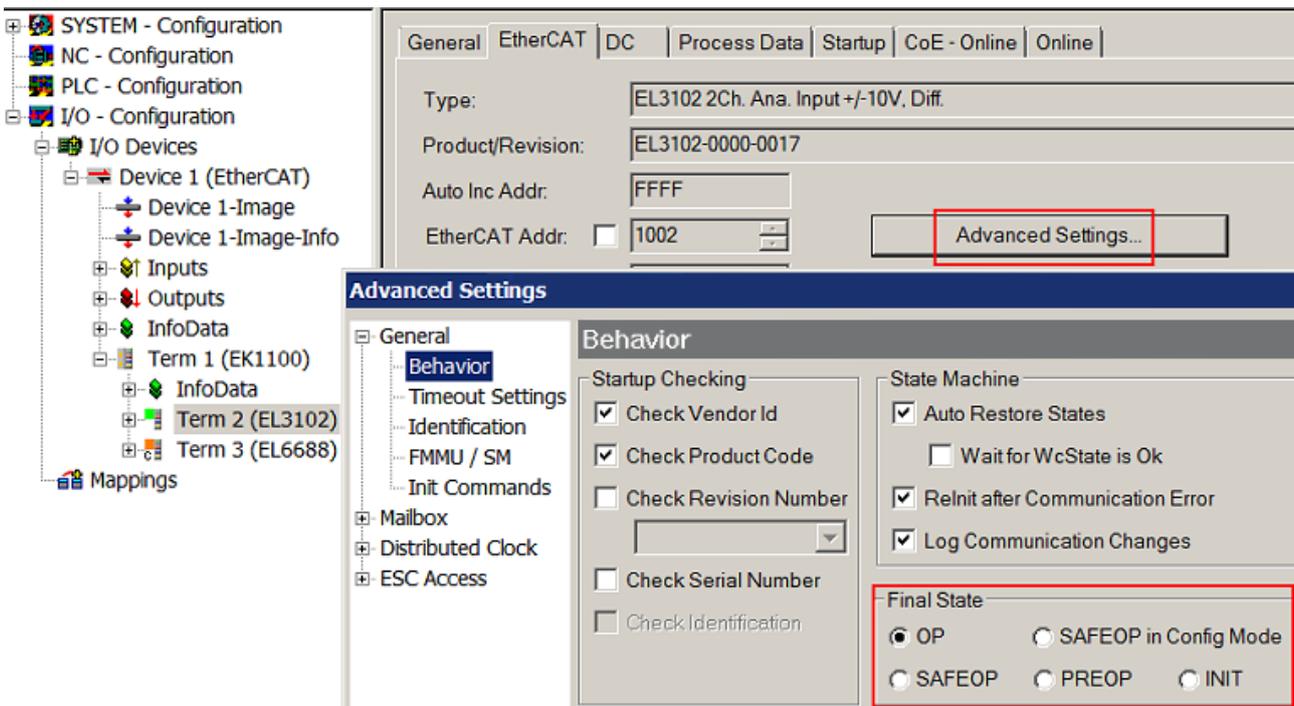


図 122: スレーブでのデフォルトのターゲット状態

手動制御

特定の理由で、アプリケーション/タスク/PLCからの状態を制御した方が良い場合があります。例：

- ・ 診断を行う場合
- ・ モーション軸の再起動制御が必要な場合
- ・ 起動タイミングの変更が望ましい場合

この場合、PLCアプリケーション内で、標準として使用可能な *TcEtherCAT.lib* から PLC ファンクションブロックを使用し、*FB_EcSetMasterState* などを使用して制御しながら状態を処理することが適切です。

この際、マスタおよびスレーブで、EtherCATマスタ内の設定をINITにすると便利です。

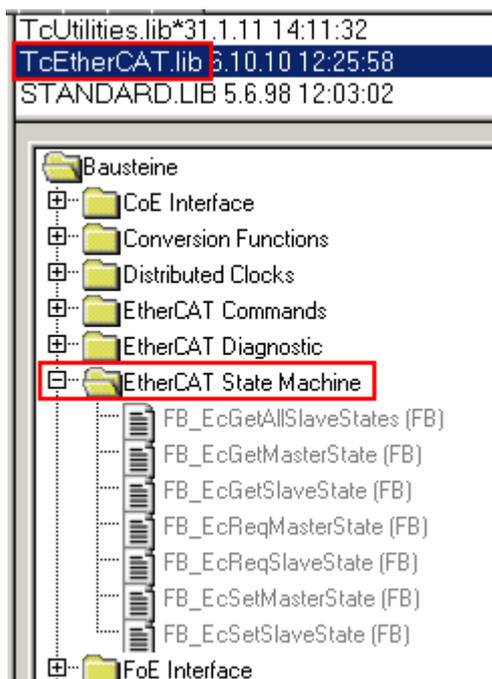


図 123: PLCファンクションブロック

Eバス電流に関する注記

EL/ESターミナルは、DINレールを使用し、カプラに続けて設置します。バスカプラは追加されたELターミナルにEバスシステム電圧5 Vを供給します。カプラは原則として最大2 Aまで供給できます。各ELターミナルがEバス電源から必要とする電流量に関する情報は、ウェブおよびカタログに記載されています。追加されたターミナルでカプラの供給上限以上の電流が必要な場合は、電源ターミナル (EL9410など) をターミナルネットワーク内の適切な個所に挿入する必要があります。

設計値として理論上の最大Eバス電流は、TwinCAT System Managerに列の値として表示されます。電流不足は負の合計量とエクスクラメーションマークで示されます。電流不足が発生する個所の前に、電源ターミナルを追加する必要があります。

General Adapter EtherCAT Online CoE - Online						
NetId:		10.43.2.149.2.1		Advanced Settings...		
Number	Box Name	Address	Type	In Size	Out S...	E-Bus (..
1	Term 1 (EK1100)	1001	EK1100			
2	Term 2 (EL3102)	1002	EL3102	8.0		1830
3	Term 4 (EL2004)	1003	EL2004		0.4	1730
4	Term 5 (EL2004)	1004	EL2004		0.4	1630
5	Term 6 (EL7031)	1005	EL7031	8.0	8.0	1510
6	Term 7 (EL2808)	1006	EL2808		1.0	1400
7	Term 8 (EL3602)	1007	EL3602	12.0		1210
8	Term 9 (EL3602)	1008	EL3602	12.0		1020
9	Term 10 (EL3602)	1009	EL3602	12.0		830
10	Term 11 (EL3602)	1010	EL3602	12.0		640
11	Term 12 (EL3602)	1011	EL3602	12.0		450
12	Term 13 (EL3602)	1012	EL3602	12.0		260
13	Term 14 (EL3602)	1013	EL3602	12.0		70
14	Term 3 (EL6688)	1014	EL6688	22.0		-240 !

図 124: Eバス電流の超過

TwinCAT 2.11以降では、コンフィグレーションが有効になっている場合、ローガーウィンドウに警告メッセージ「E-Bus Power of Terminal...」が出力されます。



図 125: 超過したEバス電流の警告メッセージ

注記
<p>注意 誤作動の可能性あり</p> <p>ターミナルブロック内のすべてのEtherCATターミナルのEバス電源に対して、同一の接地電位を使用する必要があります。</p>

5.4 CoEオブジェクトの概要

5.4.1 標準CoEオブジェクト

標準CoEオブジェクト

インデックス(16進数)が1000~1018:04のオブジェクトは、標準CoEオブジェクトとして参照されます。これらのオブジェクトには、デバイス名、ソフトウェアおよびハードウェアバージョン、メーカー固有のID(シリアル番号、リビジョン、ベンダID、および製品コード)といった、各デバイスおよび各ターミナルに関する全般情報が含まれています。

インデックス(16進数)	名前	意味	データ型	フラグ	デフォルト値
1000:0	Device type	EtherCATスレーブのデバイスタイプ: 下位ワードには、使用するCoEプロファイル(5001)が含まれます。上位ワードには、モジュール式デバイスプロファイルに基づいたモジュールプロファイルが含まれます。	UINT32	RO	0x00001389 (5001 _{dec})

インデックス(16進数)	名前	意味	データ型	フラグ	デフォルト値
1008:0	Device name	EtherCATスレーブのデバイス名	STRING	RO	EL6695-0000

インデックス(16進数)	名前	意味	データ型	フラグ	デフォルト値
1009:0	ハードウェアバージョン	EtherCATスレーブのハードウェアバージョン	STRING	RO	00

インデックス(16進数)	名前	意味	データ型	フラグ	デフォルト値
100A:0	Software version	EtherCATスレーブのファームウェアバージョン	STRING	RO	02

インデックス(16進数)	名前	意味	データ型	フラグ	デフォルト値
1018:0	Identity	スレーブ識別情報	UINT8	RO	0x04 (4 _{dec})
1018:01	Vendor ID	EtherCATスレーブのベンダID	UINT32	RO	0x00000002 (2 _{dec})
1018:02	Product code	EtherCATスレーブの製品コード	UINT32	RO	0x1A243052 (438579282 _{dec})
1018:03	Revision	EtherCATスレーブのリビジョン番号: 下位ワード(ビット0~15)は特殊ターミナルの番号を示し、上位ワード(ビット16~31)はデバイス記述ファイルを参照します。	UINT32	RO	0x00000064 (100 _{dec})
1018:04	Serial number	EtherCATスレーブのシリアル番号: 下位ワードの下位バイト(ビット0~7)には製造年が含まれ、下位ワードの上位バイト(ビット8~15)には製造された週が含まれます。上位ワード(ビット16~31)は0です。	UINT32	RO	0x00000000 (0 _{dec})

5.4.2 ターミナル固有のCoEオブジェクト

ターミナル固有のCoEオブジェクト

EL6695ブリッジターミナルのターミナル固有のCoEオブジェクトは、インデックス(16進数) 0x10F4から始まります。オブジェクト0x10F4および0x10F5には、外部同期のステータスおよび設定が含まれます。

インデックス(16進数)	名前	意味	データ型	フラグ	デフォルト値
10F4:0	External synchronization status	同期ステータスに関する情報	UINT8	RO	0x13 (19 _{dec})
10F4:01	Sync Mode	同期モード 0 = 同期なし 1 = セカンダリが同期マスタ 2 = プライマリが同期マスタ	BIT2	RO P	0x00 (0 _{dec})
10F4:0E	Control value update toggle	新しい制御値が使用可能な場合、ビットが切り替わります。	BOOLEAN	RO P	0x00 (0 _{dec})
10F4:0F	Time stamp update toggle	新しいDCデータを供給すると、ビットが切り替わります。	BOOLEAN	RO P	0x00 (0 _{dec})
10F4:10	External device not connected	0 = 他方がEtherCATフィールドバスに接続されている 1 = 他方がEtherCATフィールドバスに接続されていない	BOOLEAN	RO P	0x00 (0 _{dec})
10F4:11	Internal time stamp	内側のディストリビュートクロック時間	UINT64	RO P	-
10F4:12	External time stamp	他方のディストリビュートクロック時間(リモート側)	UINT64	RO P	-
10F4:13	Control Value for DC Master Clock	優先度の低いリファレンスクロック補正用のオフセット	INT32	RO P	0x00000000 (0 _{dec})

インデックス(16進数)	名前	意味	データ型	フラグ	デフォルト値
10F5:0	External synchronization settings	EtherCATブリッジの同期設定	UINT8	RO	0x12 (18 _{dec})
10F5:01	Sync master	0: プライマリに同期マスタがある 1: セカンダリに同期マスタがある	BOOLEAN	RW (PREOP)	0x00 (0 _{dec})
10F5:02	32 Bit time stamps	0: 64ビットタイムスタンプ 1: 32ビットタイムスタンプ	BOOLEAN	RW	0x00 (0 _{dec})
10F5:11	Control Interval (ms)	制御値計算の間隔(単位ms)	UINT16	RW	0x0000 (0 _{dec})
10F5:12	Additional System Time	制御値計算の追加DC時間	UINT64	RW	0x0000000000000000 (0 _{dec})

インデックス(16進数)	名前	意味	データ型	フラグ	デフォルト値
1608:0	RxPDO-Map	PDO Mapping RxPDO 1 (宣言されている出力プロセスデータのPDO Mapping)	UINT8	RW	0x00 (0 _{dec})
.. 1608:FF	-	-	-	-	-
...					
161F:00...					
161F:FF					

インデックス(16進数)	名前	意味	データ型	フラグ	デフォルト値
1801:0	TxPDO-Par External Sync Compact	PDOパラメータTxPDO 2	UINT8	RO	0x09 (9 _{dec})
1801:06	Exclude TxPDOs	TxPDO 2と共に転送してはならないTxPDO (TxPDO Mappingのインデックス(16進数))	OCTET-STRING[4]	RO	02 1A 03 1A
1801:07	TxPDO State	関連する入力データ内で正しく読み取りできなかった場合、TxPDO Stateがセットされます。	BOOLEAN	RO P	0x00 (0 _{dec})
1801:09	TxPDO Toggle	対応する入力データを更新するたびに、TxPDO Toggleが切り替わります。	BOOLEAN	RO P	0x00 (0 _{dec})

インデックス(16進数)	名前	意味	データ型	フラグ	デフォルト値
1802:0	TxPDO-Par External Sync	PDOパラメータTxPDO 3	UINT8	RO	0x09 (9 _{dec})
1802:06	Exclude TxPDOs	TxPDO 3と共に転送してはならないTxPDO (TxPDO Mappingオブジェクトのインデックス(16進数))	OCTET-STRING[4]	RO	01 1A 03 1A
1802:07	TxPDO State	関連する入力データ内で正しく読み取りできなかった場合、TxPDO Stateがセットされます。	BOOLEAN	RO P	0x00 (0 _{dec})
1802:09	TxPDO Toggle	対応する入力データを更新するたびに、TxPDO Toggleが切り替わります。	BOOLEAN	RO P	0x00 (0 _{dec})

インデックス(16進数)	名前	意味	データ型	フラグ	デフォルト値
1803:0	TxPDO-Par External Sync (32 Bit)	PDOパラメータTxPDO 4	UINT8	RO	0x09 (9 _{dec})
1803:06	Exclude TxPDOs	TxPDO 4と共に転送してはならないTxPDO (TxPDO Mappingオブジェクトのインデックス(16進数))	OCTET-STRING[4]	RO	01 1A 02 1A
1803:07	TxPDO State	関連する入力データ内で正しく読み取りできなかった場合、TxPDO Stateがセットされます。	BOOLEAN	RO P	0x00 (0 _{dec})
1803:09	TxPDO Toggle	対応する入力データを更新するたびに、TxPDO Toggleが切り替わります。	BOOLEAN	RO P	0x00 (0 _{dec})

インデックス(16進数)	名前	意味	データ型	フラグ	デフォルト値
1A01:0	TxPDO-Map External Sync Compact	PDO Mapping TxPDO 1	UINT8	RW	0x05 (5 _{dec})
1A01:01	SubIndex (hex) 001	12ビットアライメント	UINT32	RW	0x0000:00, 12
1A01:02	SubIndex (hex) 002	PDO Mapping エントリ(オブジェクト0x1801、エントリ0x09)	UINT32	RW	0x1801:09, 1
1A01:03	SubIndex (hex) 003	PDO Mapping エントリ(オブジェクト0x1801、エントリ0x07)	UINT32	RW	0x1801:07, 1
1A01:04	SubIndex (hex) 004	1ビットアライメント	UINT32	RW	0x0000:00, 1

インデックス(16進数)	名前	意味	データ型	フラグ	デフォルト値
1A01:05	SubIndex (hex) 005	PDO Mapping エントリ(オブジェクト0x10F4 (外部同期ステータス)、エントリ0x10)	UINT32	RW	0x10F4:10, 1

インデックス(16進数)	名前	意味	データ型	フラグ	デフォルト値
1A02:0	TxPDO-Map External Sync	PDO Mapping TxPDO 2	UINT8	RW	0x09 (9 _{dec})
1A02:01	SubIndex (hex) 001	PDO Mapping エントリ(オブジェクト0x10F4 (外部同期ステータス)、エントリ0x02)	UINT32	RW	0x10F4:01, 2
1A02:02	SubIndex (hex) 002	10ビットアライメント	UINT32	RW	0x0000:00, 10
1A02:03	SubIndex (hex) 003	PDO Mapping エントリ(オブジェクト0x1802、エントリ0x09)	UINT32	RW	0x1802:09, 1
1A02:04	SubIndex (hex) 004	PDO Mapping エントリ(オブジェクト0x1802、エントリ0x07)	UINT32	RW	0x1802:07, 1
1A02:05	SubIndex (hex) 005	PDO Mapping エントリ(オブジェクト0x10F4 (外部同期ステータス)、エントリ0x0F (タイムスタンプ更新トグル))	UINT32	RW	0x10F4:0F, 1
1A02:06	SubIndex (hex) 006	PDO Mapping エントリ(オブジェクト0x10F4 (外部同期ステータス)、エントリ0x10 (外部デバイス接続なし))	UINT32	RW	0x10F4:10, 1
1A02:07	SubIndex (hex) 007	PDO Mapping エントリ(オブジェクト0x10F4 (外部同期ステータス)、エントリ0x11 (内部タイムスタンプ))	UINT32	RW	0x10F4:11, 64
1A02:08	SubIndex (hex) 008	PDO Mapping エントリ(オブジェクト0x10F4 (外部同期ステータス)、エントリ0x12 (外部タイムスタンプ))	UINT32	RW	0x10F4:12, 64
1A02:09	SubIndex (hex) 009	PDO Mapping エントリ(オブジェクト0x10F4 (外部同期ステータス)、エントリ0x13 (予約))	UINT32	RW	0x10F4:13, 32

インデックス(16進数)	名前	意味	データ型	フラグ	デフォルト値
1A03:0	TxPDO-Map External Sync	PDO Mapping TxPDO 3	UINT8	RW	0x09 (9 _{dec})
1A03:01	SubIndex (hex) 001	PDO Mapping エントリ(オブジェクト0x10F4 (外部同期ステータス)、エントリ0x02)	UINT32	RW	0x10F4:01, 2
1A03:02	SubIndex (hex) 002	10ビットアライメント	UINT32	RW	0x0000:00, 10
1A03:03	SubIndex (hex) 003	PDO Mapping エントリ(オブジェクト0x1803、エントリ0x09)	UINT32	RW	0x1803:09, 1
1A03:04	SubIndex (hex) 004	PDO Mapping エントリ(オブジェクト0x1803、エントリ0x07)	UINT32	RW	0x1803:07, 1
1A03:05	SubIndex (hex) 005	PDO Mapping エントリ(オブジェクト0x10F4 (外部同期ステータス)、エントリ0x0F (タイムスタンプ更新トグル))	UINT32	RW	0x10F4:0F, 1
1A03:06	SubIndex (hex) 006	PDO Mapping エントリ(オブジェクト0x10F4 (外部同期ステータス)、エントリ0x10 (外部デバイス接続なし))	UINT32	RW	0x10F4:10, 1

インデックス(16進数)	名前	意味	データ型	フラグ	デフォルト値
1A03:07	SubIndex (hex) 007	PDO Mapping エントリ(オブジェクト0x10F4 (外部同期ステータス)、エントリ0x11 (内部タイムスタンプ))	UINT32	RW	0x10F4:11, 32
1A03:08	SubIndex (hex) 008	PDO Mapping エントリ(オブジェクト0x10F4 (外部同期ステータス)、エントリ0x12 (外部タイムスタンプ))	UINT32	RW	0x10F4:12, 32
1A03:09	SubIndex (hex) 009	PDO Mapping エントリ(オブジェクト0x10F4 (外部同期ステータス)、エントリ0x13 (予約))	UINT32	RW	0x10F4:13, 32

インデックス(16進数)	名前	意味	データ型	フラグ	デフォルト値
1A04:0	Active TxPDOs-Map	PDO Mapping TxPDO 4	UINT32	RW	0x05 (5 _{dec})
1A04:01	SubIndex (hex) 01	PDO Mapping エントリ(オブジェクト0xF640:01(リモート書き込みサイクルu16Count))	UINT32	RW	10 01 40 F6
1A04:02	SubIndex (hex) 02	PDO Mapping エントリ(オブジェクト0xF630:01 Active TxPdo Info PDO 1-8)	UINT32	RW	10 01 30 F6
1A04:03	SubIndex (hex) 03	PDO Mapping エントリ(オブジェクト0xF630:02 Active TxPdo Info PDO 9-16)	UINT32	RW	10 02 30 F6
1A04:04	SubIndex (hex) 04	PDO Mapping エントリ(オブジェクト0xF630:03 Active TxPdo Info PDO 17-24)	UINT32	RW	10 03 30 F6
1A04:05	SubIndex (hex) 05	PDO Mapping エントリ(オブジェクト0xF630:04 Active TxPdo Info PDO 25-32)	UINT32	RW	10 04 30 F6

インデックス(16進数)	名前	意味	データ型	フラグ	デフォルト値
1A05:0	FoE Info-Map	PDO Mapping TxPDO 5	UINT8	RW	0x01 (1 _{dec})
1A05:01		PDO Mapping エントリ(オブジェクト0xF650:01 Active TxPdo Info PDO 25-32)	UINT32	RW	10 01 50 F6

インデックス(16進数)	名前	意味	データ型	フラグ	デフォルト値
1A08:0	TxPDO-Map	PDO Mapping TxPDO 1 (宣言されている入力プロセスデータのPDO Mapping)	UINT8	RW	0x00 (0 _{dec})
.. 1A08:FF	-	-	-	-	-
...					
1A1F:00...					
1A1F:FF					

インデックス(16進数)	名前	意味	データ型	フラグ	デフォルト値
1C00:0	Syncmanager type	Sync Managerタイプチャンネル(メールボックス/プロセスデータ、読み取り/書き込み)	UINT8	RO	0x04 (4 _{dec})
1C00:01	SubIndex (hex) 001	Sync Managerタイプチャンネル1:メールボックス書き込み	UINT8	RO	0x01 (1 _{dec})
1C00:02	SubIndex (hex) 002	Sync Managerタイプチャンネル2:メールボックス読み取り	UINT8	RO	0x02 (2 _{dec})
1C00:03	SubIndex (hex) 003	Sync Managerタイプチャンネル3:プロセスデータ書き込み(出力)	UINT8	RO	0x03 (3 _{dec})

インデックス(16進数)	名前	意味	データ型	フラグ	デフォルト値
1C00:04	SubIndex (hex) 004	Sync Managerタイプチャンネル4: プロセスデータ読み取り(入力)	UINT8	RO	0x04 (4 _{dec})

インデックス(16進数)	名前	意味	データ型	フラグ	デフォルト値
1C12:0	RxPDO assign	RxPDO assign出力	UINT8	RW	0x00 (0 _{dec})
1C32:01	SubIndex (hex) 001	割り当てられた1番目のRxPDO (関連するRxPDO Mappingオブジェクトのインデックス(16進数)を含む)	UINT16	RW	0x0000 (0 _{dec})

インデックス(16進数)	名前	意味	データ型	フラグ	デフォルト値
1C13:0	TxPDO assign	PDO assign入力	UINT8	RW	0x01 (1 _{dec})
1C13:01	SubIndex (hex) 001	割り当てられた1番目のTxPDO (関連するTxPDO Mappingオブジェクトのインデックス(16進数)を含む)	UINT16	RW	0x1A01 (6657 _{dec})
1C13:02	SubIndex (hex) 002	割り当てられた2番目のTxPDO (関連するTxPDO Mappingオブジェクトのインデックス(16進数)を含む)	UINT16	RW	0x0000 (0 _{dec})

インデックス(16進数)	名前	意味	データ型	フラグ	デフォルト値
1C32:0	SM output parameter	出力の同期パラメータ	UINT8	RO	0x20 (32 _{dec})
1C32:01	Sync mode	現在の同期モード: 0: FreeRun 1: SM 2イベントで同期 2: DC-Mode - SYNC0イベントで同期 3: DC-Mode - SYNC1イベントで同期	UINT16	RW	0x0000 (0 _{dec})
1C32:02	Cycle time	サイクルタイム(単位ns): FreeRun: ローカルタイマのサイクルタイム SM 2イベントで同期: マスタサイクルタイム DC-Mode: SYNC0/SYNC1サイクルタイム	UINT32	RW	0x00000000 (0 _{dec})
1C32:03	Shift time	SYNC0イベントから出力までの時間(単位ns、DCモードのみ)	UINT32	RW	0x00000000 (0 _{dec})
1C32:04	Sync modes supported	サポートしている同期モード: ビット0 = 1: FreeRunをサポート ビット1 = 1: SM 2イベントで同期をサポート ビット2-3 = 01: DCモードをサポート ビット4-5 = 10: SYNC1イベントでの出力シフト(DCモードのみ) ビット14 = 1: 動的回数(1C32:08の書き込みによる計測)	UINT16	RO	0xC007 (49159 _{dec})
1C32:05	Minimum cycle time	最小サイクルタイム(単位ns)	UINT32	RO	0x00000000 (0 _{dec})

インデックス(16進数)	名前	意味	データ型	フラグ	デフォルト値
1C32:06	Calc and copy time	SYNC0からSYNC1イベントまでの最小時間(単位ns、DCモードのみ)	UINT32	RO	0x00000000 (0 _{dec})
1C32:08	コマンド	0: ローカルサイクルタイムの計測を停止 1: ローカルサイクルタイムの計測を開始 エントリ1C32:03、1C32:05、1C32:06、1C32:09、1C33:03、1C33:06、および1C33:09は、最大測定値で更新されます。後続の計測のために、測定値がリセットされます。	UINT16	RW	0x0000 (0 _{dec})
1C32:09	Maximum delay time	SYNC1イベントから出力までの時間(単位ns、DCモードのみ)	UINT32	RO	0x00000000 (0 _{dec})
1C32:0B	SM event missed counter	OPERATIONALでのSMイベントの欠損数(DCモードのみ)	UINT16	RO	0x0000 (0 _{dec})
1C32:0C	Cycle exceeded counter	OPERATIONALでサイクル時間を超過した回数(サイクルが時間内に完了しなかった、または次のサイクルの開始が早すぎた場合)	UINT16	RO	0x0000 (0 _{dec})
1C32:0D	Shift too short counter	SYNC0とSYNC1イベント間の間隔が短すぎた回数(DCモードのみ)	UINT16	RO	0x0000 (0 _{dec})
1C32:20	Sync error	最後のサイクルで正常に同期されなかった(出力が遅すぎた、DCモードのみ)	BOOLEAN	RO	0x00 (0 _{dec})

インデックス(16進数)	名前	意味	データ型	フラグ	デフォルト値
1C33:0	SM input parameter	入力の同期パラメータ	UINT8	RO	0x20 (32 _{dec})
1C33:01	Sync mode	現在の同期モード: 0: FreeRun 1: SM 3イベントで同期(出力PDOがない場合) 2: DC - SYNC0イベントで同期 3: DC - SYNC1イベントで同期 34: SM 2イベントで同期(出力PDOがある場合)	UINT16	RW	0x0000 (0 _{dec})
1C33:02	Cycle time	1C32:02と同様	UINT32	RW	0x00000000 (0 _{dec})
1C33:03	Shift time	SYNC0イベントから入力の読み取りまでの時間(単位ns、DCモードのみ)	UINT32	RW	0x00000000 (0 _{dec})

インデックス(16進数)	名前	意味	データ型	フラグ	デフォルト値
1C33:04	Sync modes supported	サポートしている同期モード: ビット0: FreeRunをサポート ビット1: SM 2イベントで同期をサポート(出力PDOがある場合) ビット1: SM 3イベントで同期をサポート(出力PDOがない場合) ビット2-3 = 01: DCモードをサポート ビット4-5 = 01: ローカルイベントによる入力シフト(出力あり) ビット4-5 = 10: SYNC1イベントで入力シフト(出力なし) ビット14 = 1: 動的回数(1C32:08または1C33:08の書き込みによる計測)	UINT16	RO	0xC007 (49159 _{dec})
1C33:05	Minimum cycle time	1C32:05と同様	UINT32	RO	0x00000000 (0 _{dec})
1C33:06	Calc and copy time	入力の読み取りからマスタの入力が使用可能になるまでの時間(単位ns、DCモードのみ)	UINT32	RO	0x00000000 (0 _{dec})
1C33:08	コマンド	1C32:08と同様	UINT16	RW	0x0000 (0 _{dec})
1C33:09	Maximum delay time	SYNC1イベントから入力の読み取りまでの時間(単位ns、DCモードのみ)	UINT32	RO	0x00000000 (0 _{dec})
1C33:0B	SM event missed counter	1C32:11と同様	UINT16	RO	0x0000 (0 _{dec})
1C33:0C	Cycle exceeded counter	1C32:12と同様	UINT16	RO	0x0000 (0 _{dec})
1C33:0D	Shift too short counter	1C32:13と同様	UINT16	RO	0x0000 (0 _{dec})
1C33:20	Sync error	1C32:32と同様	BOOLEAN	RO	0x00 (0 _{dec})

5.4.3 プロファイル固有のCoEオブジェクト

● CoEオブジェクト0x6000および0x7000の読み取り

i CoEオブジェクト0x6000および0x7000の読み取り中に出力されるデータは、実際のプロセスデータではありません。実際のデータは、PDO割り当てから間接的に読み取ることしかできません。

インデックス(16進数)	名前	意味	データ型	フラグ	デフォルト値
6000:0	Input Data	宣言された入力プロセスデータ(動的に作成)	UINT8	RO	0x00 (0 _{dec})
6000:01					
...					
6000:FF					

インデックス(16進数)	名前	意味	データ型	フラグ	デフォルト値
7000:0	Output Data	宣言された出力プロセスデータ(動的に作成)	UINT8	RO	0x00 (0 _{dec})
7000:01					
...					
7000:FF					

インデックス (16進数)	名前	意味	データ型	フラグ	デフォルト値
F000:0	Modular device profile	モジュール型デバイスプロファイルの全般情報	UINT8	R0	0x02 (2 _{dec})
F000:01	Module Index (hex) distance	個々のチャンネルのオブジェクトのインデックス (16進数) 間隔	UINT16	R0	0x0010 (16 _{dec})
F000:02	Maximum number of modules	チャンネル数	UINT16	R0	0x0001 (1 _{dec})

インデックス (16進数)	名前	意味	データ型	フラグ	デフォルト値
F008:0	Code word	現状は予備	UINT32	RW	0x00000000 (0 _{dec})

インデックス (16進数)	名前	意味	データ型	フラグ	デフォルト値
F010:0	Module list	最大サブインデックス (16進数)	UINT8	RW	0x01 (1 _{dec})
F010:01	SubIndex (hex) 001	-	UINT32	RW	0x00000000 (0 _{dec})

現状は64個のPDO、1つのTx-PDOにつき2ビット

ビット0 → 1=マッピングあり (p)

ビット1 → 1=マッピングが有効 (a)

インデックス (16進数)	名前
F630	Active TxPdo
マッピング:	a p a p a p a p a p a p a p a p
0	0
1	1A07 1A06 1A05 1A04 1A03 1A02 1A01 1A00
2	1A0F 1A0E 1A0D 1A0C 1A0B 1A0A 1A09 1A08
3	1A17 1A16 1A15 1A14 1A13 1A12 1A11 1A10
4	1A1F 1A1E 1A1D 1A1C 1A1B 1A1A 1A19 1A18
...
8	1A3F 1A3E 1A3D 1A3C 1A3B 1A3A 1A39 1A38

インデックス (16進数)	ビット	名前	意味	コメント
F800:0	-	Device Config		
F800:01	0x8000	AoE	1 = プロトコルが無効	
	0x4000	EoE		
	0x2000	FoE		
	0x1000	SoE		
	0x0800	VoE		
	0x0400	その他		
	0x0200	-		
	...			
	0x0002	-		エラーメッセージ抑制
	0x0001	-		1: ルートされていない場合: エラーメッセージ抑制
	...			
0xF800:02	0x0100	FoE Buffer	Enable	
	...			

インデックス(16進数)	ビット	名前	意味	コメント
	0x0002	OP State	他方への依存性を無効化	セカンダリの動作状態への依存性を無効化(デバイスエミュレーションのみ)
	0x0001	OP State		プライマリの動作状態への依存性を無効化(デバイスエミュレーションのみ)
0xF800:03	0x2000		SW再起動	AnyからInit (A → I)への状態変化中に再起動
	...			

インデックス(16進数)	名前	意味	データ型	フラグ	デフォルト値
F820:0	ADS Server Settings	宛先Net ID/ポート	Object	RW	0x02 (2 _{dec})
F820:01	Net ID	宛先Net ID	Array [0..5]	RW	0x00, ..
F820:02	Port	宛先ポート	UINT16	RW	0x0000

インデックス(16進数)	名前	意味	データ型	フラグ	デフォルト値
F821:0	EL6695 ADS Settings	ソースNet ID/ポート	Object	RO	0x02 (2 _{dec})
F821:01	Net ID	ソースNet ID	Array [0..5]	RO	0x00, ..
F821:02	Port	ソースポート	UINT16	RO	0x0000

インデックス(16進数)	名前	意味	データ型	フラグ	デフォルト値
FA20:0	Device Diag	ステータス、GPU、およびヒープロードに関する診断情報、送信データパケットおよび内部コピー時間に関する情報を含む。	UINT16	RD	0x1D (29 _{dec})
FA20:01	Status		UINT16	RD	
FA20:02	CPU usage [%]		UINT16	RD	
FA20:03	Heap Usage [%]		UINT16	RD	
FA20:04	AOE Packets		UINT16	RD	
FA20:05	EOE Packets		UINT16	RD	
FA20:06	FOE Packets		UINT16	RD	
FA20:07	SOE Packets		UINT16	RD	
FA20:08	VOE Packets		UINT16	RD	
FA20:09	Other Packets		UINT16	RD	
FA20:0A	Mbx Info		UINT16	RD	
FA20:0B	PD Copy time (my)	時間(単位μs)	UINT16	RD	
FA20:0C	PD Copy time (remote)		UINT16	RD	
FA20:0D..	Info 2		UINT16	RD	
FA20:1D	Info 18		UINT16	RD	

6 機能および動作モード

ELターミナルの動作モードは、プロセスデータ設定に基づいて基本的動作の原則を定義します。

6.1 基本機能の原則

EL6695 EtherCATブリッジターミナルは、EtherCATネットワークと2つの異なるマスタ間の同期リアルタイムデータ交換を可能にします。AoE、FoE、EoE、VoEなどの各種非周期的プロトコルによる非同期通信もサポートしています。

EL6695には、以下の3つの基本機能が用意されています。

- ・ 周期PDOデータ交換
- ・ 非周期データ交換
- ・ ディストリビュートクロック同期

周期プロセスデータの同期データ交換中に、EtherCATマスタは事前設定したプロセスデータを一方のブリッジ側から他方にコピーします。両側のステータス変数は、欠落データおよび他方のステータスに関する情報を提供します。

ディストリビュートクロックは双方向で同期できます。どちらの側に上位のリファレンスクロック（いわゆるグランドマスタクロック）が含まれているかを定義することが重要です。EL6695は他方の従属EtherCATマスタに時間値を転送し、従属EtherCATマスタがこの機能をサポートしていれば、自身のEtherCATシステムタイムを調整できます。

EL6695はEL6692にはなかった、CoEコンフィグレーションの柔軟性、デバイスエミュレーションオプション、およびデータスループットの大幅な向上といった拡張機能を搭載しています（EL6692も引き続きお求めいただけます）。

TwinCAT 3のTwinCAT System Managerには、ユーザフレンドリなコンフィグレーションインターフェイスが用意されています（TwinCAT 2では準備中）。このエクステンションはEmulationModeで必要となりますが、一般的な操作にも欠かせません。

EL6695の内部電源は、プライマリ（Eバス）またはセカンダリ（RJ45）から冗長的に供給されます。これらの2つのどちらか一方からでも供給されると、EL6695は両方のEtherCATで完全に機能します。プライマリにはEバス経由で、セカンダリ（RJ45）には外部接続経由で供給されます。両方の電源が有効な場合、24 V電源が優先的に使用されます。

EL6695は、1つの筐体内に2つの完全なEtherCATスレーブを効果的に内蔵しています。1つはプライマリ（Eバス）のEL6695-0000、もう1つはネットワークケーブル接続（RJ45）によるセカンダリのEL6695-0002です。

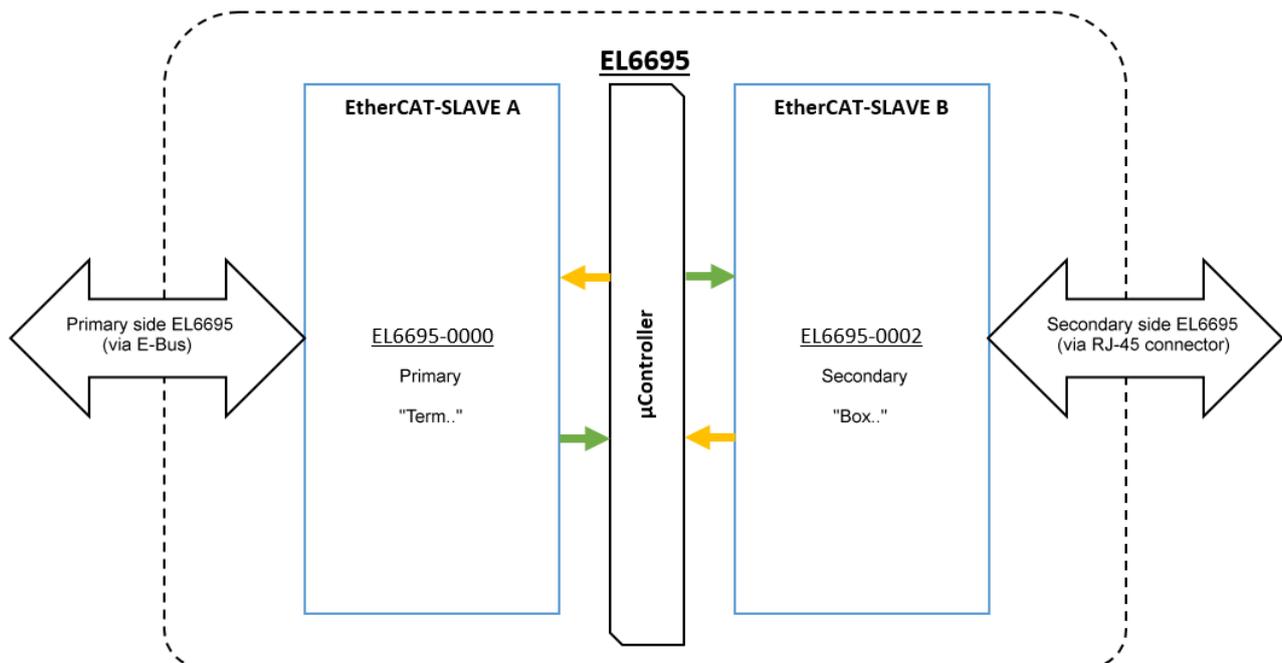


図 126: ブリッジの基本機能

機能的には、ブリッジの両側で機能は同一です。つまり、すべての機能を両側で動作または使用できます。EL6695の「プライマリ」、「セカンダリ」には優劣は存在しませんが、便宜上、Eバスに対し「プライマリ」、RJ45に対し「セカンダリ」という用語を引き続き使用します。

6.1.1 TwinCATでのEL6695へのアクセス

コンテキストメニュー(右クリック後、[Add New Item...]を選択)を使用して、コンフィグレーションの作成、およびTwinCATプロジェクト内へのEL6695ブリッジターミナルを追加できます。または、該当するEtherCATデバイスを右クリックし、デバイスn (EtherCAT)のコンテキストメニューからスキャン機能を選択すると、接続されているすべてのデバイスおよびボックスが自動的に読み込まれ、構造内に追加されます(図を参照)。

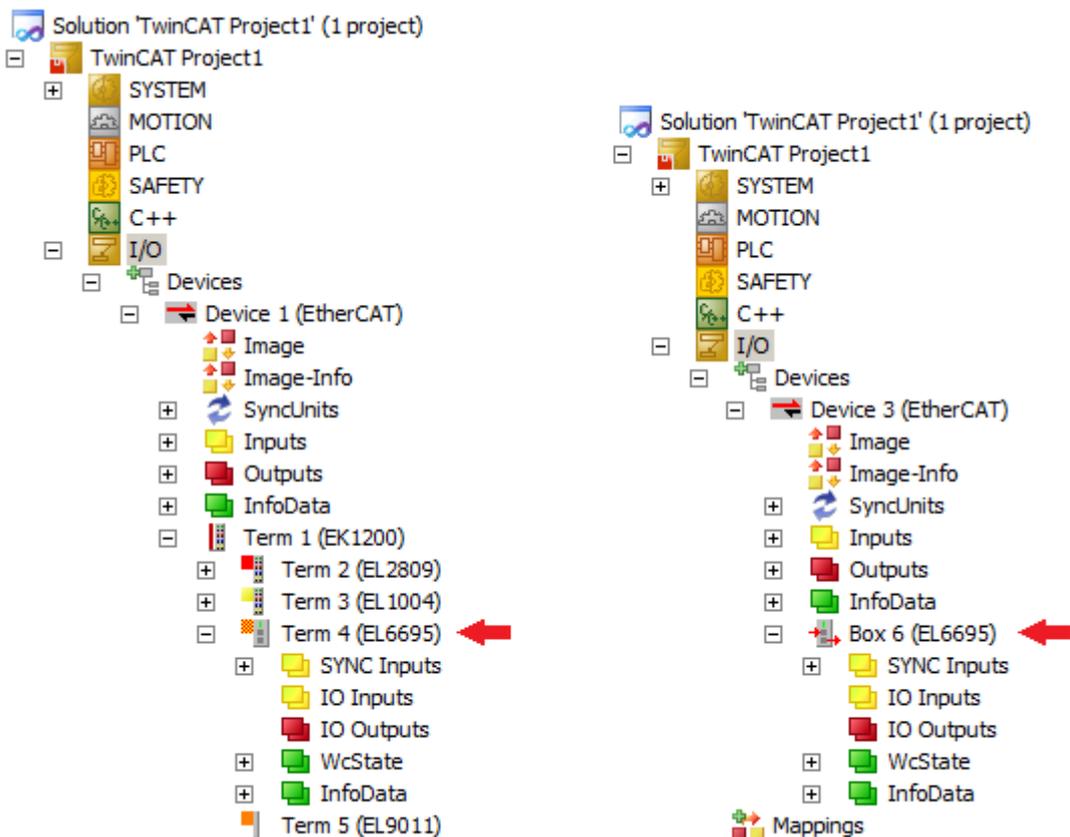


図 127: 「スキャン」後: ターミナルセグメント内のEL6695(左)、EtherCAT経由(RJ-45 X1経由)のボックス(右)

Eバス側では、EL6695はターミナルセグメント内に「Terminal」として表示されます。RJ45接続経由の場合、コンフィグレーション内に「Box」として表示されます(参考:「EtherCATスレーブ」)。スキャン/追加後のEL6695ブリッジターミナルの初期状態は、以下のとおりです。

- ・ FreeRun => DCサポート無効
- ・ 事前設定のプロセスデータ変数なし
- ・ ディストリビュートクロック情報なし

6.2 EL6692との互換性

EL6695はEL6692と互換性のある動作モードをサポートしており、その場合は同一のデフォルトプロセスデータ(ステータス/診断)を使用していますが、基本的には新しい機種とみなす必要があります。時間特性が異なるため、EL6695はEL6692と完全には互換性がないことに注意してください。

0x1A02または0x1A03内の診断データを選択すると、SyncMode、TxPdoToggle、TxPDO State、「Timestamp update toggle」および「External device not connected」などのEL6692と同様の診断データを表示できます。

Name	Typ	Grö...	>Adr...	Ein/...	User.
◆↑ Sync Mode	BIT2	0.2	39.0	Eing...	0
◆↑ TxPDO toggle	BOOL	0.1	40.4	Eing...	0
◆↑ TxPDO state	BOOL	0.1	40.5	Eing...	0
◆↑ Timestamp update toggle	BOOL	0.1	40.6	Eing...	0
◆↑ External device not connected	BOOL	0.1	40.7	Eing...	0

図 128: EL6695互換モード(デフォルトPDO)

6.3 ステートマシンEL6695

EL6695ブリッジターミナルは、両側(プライマリおよびセカンダリ)で次のEtherCAT状態をサポートしています: INIT、PreOP、SafeOP、OP、BOOTSTRAP。

片方でのステータス変更は、他方のステータスには影響を与えません。EmulationModeの場合は、この限りではありません。EmulationModeに関する説明をご参照ください。

● INIT状態

EL6695ブリッジターミナルは、片側での「通常」のINIT状態リクエストを他方に影響を与えずに処理します。基本的なターミナル設定(FW更新、デバイスエミュレーションの変更、部分的なオブジェクトディレクトリの変更など)の変更後、ターミナルは両側で再起動します。

6.4 周期プロセスデータPDO

違いを明確にするため、EL6695では次の2つの用語が使用されます。

- ・ 「対称PDO Mapping」: サイズおよび順序が同一なプロセスデータを両側で作成
- ・ 「選択的PDO Mapping」: 片側で最大量のプロセスデータをEL6695にロードし、そこから他方のマスタがサブセットを選択して、そのサブセットを周期的プロセスデータとしてリクエスト

主要データ(2015年6月時点、FW04、TwinCAT 3.1 b4018.4)

- 各方向で最大255の変数(PDOを手動作成する場合、TwinCATによって0x1A08:nnおよび0x1608:nn内の最大255のエントリを管理)。
- 最大合計PDOサイズ: MTU (~1500バイト)、1方向の1イーサネットフレームにつき約1408バイトのユーザデータに対応。必要に応じてより大きなサイズも使用可能。その場合、TwinCAT内での調整が必要。
- 最小PDOサイズ1バイト、ビット-PDOは使用不可。

コンフィグレーションを変更しても、対応するEL6695のプライマリまたはセカンダリが再起動するまでは反映されません。

0x6000番台の各オブジェクトに対して、システムは他方(プライマリまたはセカンダリ)で対応する0x7000番台のオブジェクトを検索します(0x6001:05 → 0x7001:05など)。他方のオブジェクト(0x7000番台)のサイズは、0x6000番台のオブジェクトのサイズ以上でなければなりません。この場合、サブセット0x7000番台を0x6000番台にコピーすることも可能です(参考: 「選択的PDO Mapping」)。

6.4.1 フロー制御

フロー制御はアプリケーションで行う必要があります。つまり、EL6695はハンドシェイクメカニズムを採用していません。ただし、どちらのPDOモードでもカウンタを表示できます。このカウンタは他方から書き込みアクセスがあるたびにインクリメントされます: CoEオブジェクト0xF640。このカウンタが周期PDOとして確実にマッピングするには、スタートアップエントリによって以下のようにマッピングする必要があります。

CoEオブジェクト0xF640をマッピングできるようにするには、[セクション \[▶ 22\]](#)内の説明のとおり、先にすべてのオンラインCoEオブジェクトを表示する必要があります。これにより、デバイスのPDOデータを[Process Data]タブでリロードでき、PDOリストにエントリ「Active TX PDOs Map」が表示されます。Sync Managerでエントリ「Inputs」が選択されている場合、[PDO Assignment]でオブジェクト0x1A04にチェックマークを付けられ、CoEオブジェクト0xF640のカウンタがプロセスイメージにマッピングされます。

General EtherCAT DC **Process Data** Startup CoE - Online Online EL6692

1

Sync Manager:

SM	Size	Type	Fla...
0	1024	Mbx...	
1	1024	MbxIn	
2	0	Outp...	
3	12	Inputs	

4

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	0.0	IO Inputs	MV	3	0
0x1A01	2.0	SYNC Inputs	F	3	0
0x1A02	22.0	SYNC Inputs	F		0
0x1A03	14.0	SYNC Inputs	F		0
0x1A04	10.0	Active TX-PDOs-Map		3	0
0x1600	0.0	IO Outputs	MV	2	0

3

PDO Assignment (0x1C13):

- 0x1A00
- 0x1A01
- 0x1A02 (excluded by 0x1A01)
- 0x1A03 (excluded by 0x1A01)
- 0x1A04

5

Download:

- PDO Assignment
- PDO Configuration

PDO Content (0x1A02):

Index	Size	Offs	Name	Type	Default (h...
0x10F4...	0.2	0.0	Sync Mode	BIT2	
---	0.6	0.2	---		
---	0.3	1.0	---		
0x1802...	0.1	1.3	TxPDO-Toggle	BIT	

Predefined PDO Assignment (none)

2

Load PDO info from device

Sync Unit Assignment...

図 129: デバイスからのPDOデータのロードによりエントリ「Active TX PDOs Map」を作成

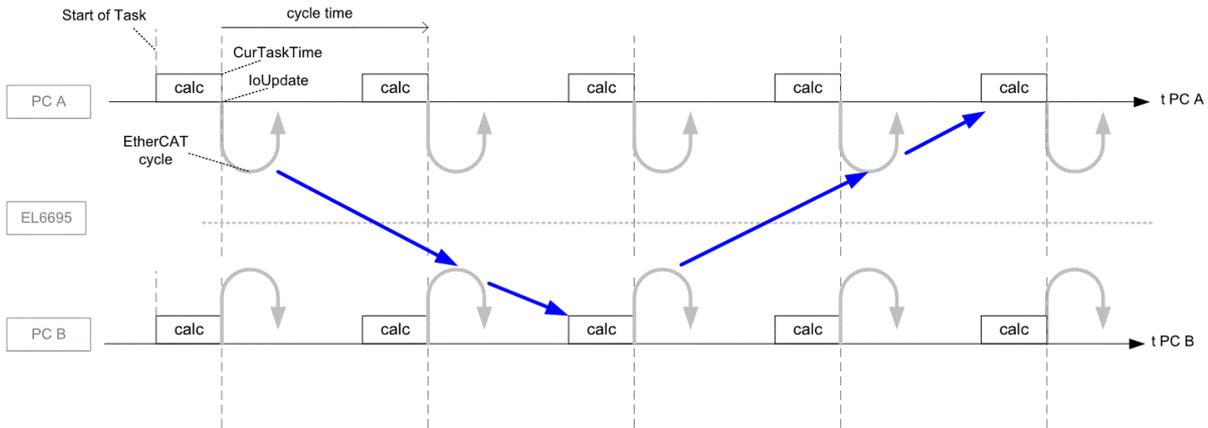
- Box 4 (EL6695)
 - IO Inputs
 - SYNC Inputs
 - Active TX-PDOs-Map
 - u16Count**
 - PDO 1-8
 - PDO 9-16
 - PDO 17-24
 - PDO 25-32
 - IO Outputs
 - WcState
 - InfoData

図 130: PDOの構造

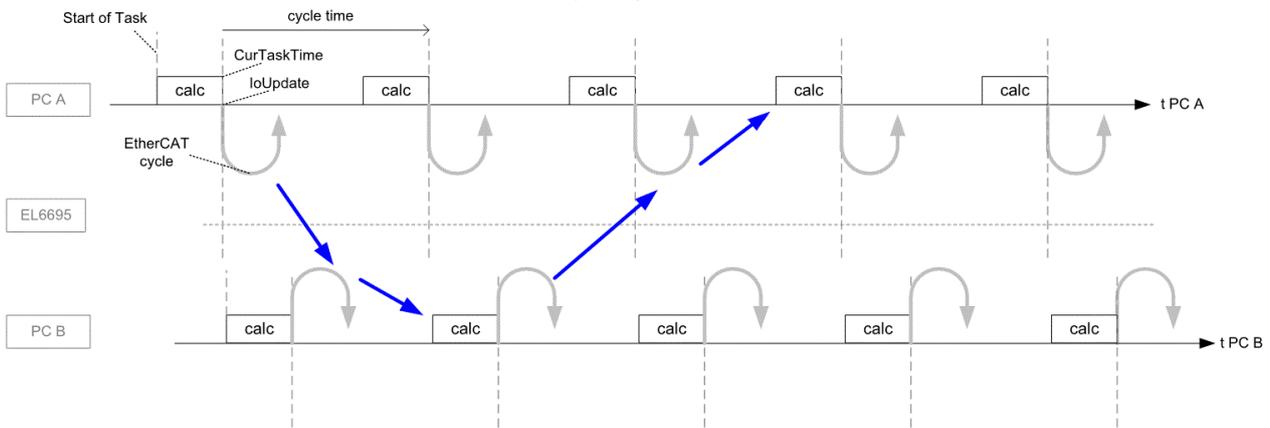
データスループット(例)

以下の図は、標準コンフィグレーションを使用した場合と、2つの最適化メソッド(同期による最適化、および個別のIO更新による最適化)を使用した場合のEL6695のデータスループット、および同期を示しています。

Standard Verhalten I/O am Taskende - durch Jitter der Tasks und Unterschiedliche Startzeitpunkte der Tasks zwischen 4 bis 6 Zyklen
 - da die EL6695 schneller als EtherCAT ist, tritt dort keine Verzögerung auf



Optimierung durch Synchronisation der Tasks, sodass Task B immer nach Task A das I/O Update durchläuft
 - somit wäre ein Unterschied von 3 Zyklen möglich



Optimierung durch separates I/O Update und Synchronisation, Inputs am Anfang Outputs am Ende
 - somit wäre ein Unterschied von 2 Zyklen möglich

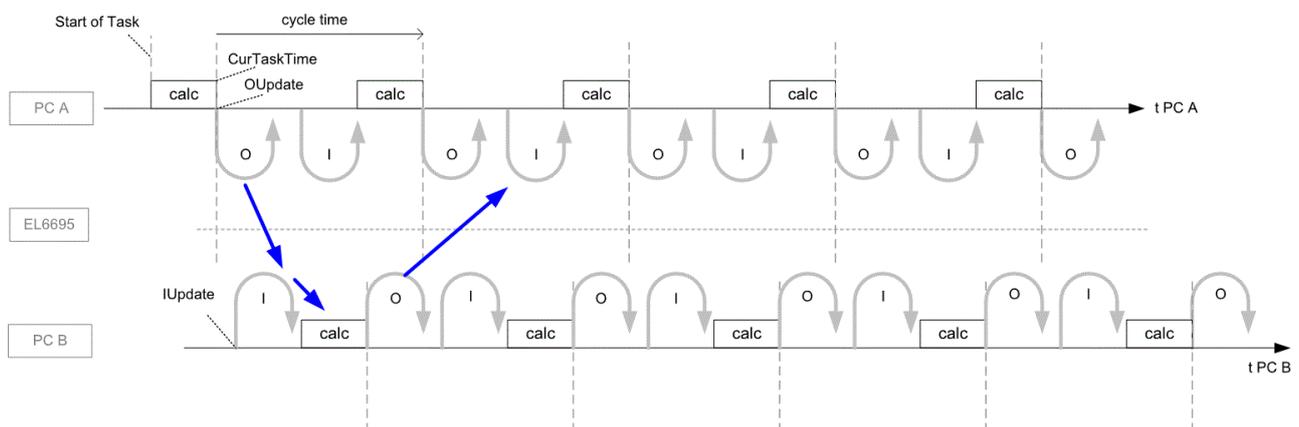


図 131: データスループット: 標準の場合と最適化を行った場合

設定したプロセスデータを一方のEtherCATから他方に転送する際にかかる時間は、データ量によって異なります。

内部データ転送は、一方での読み取りによってトリガされます。以下の図は、このシーケンスを示しています。

- SyncManager 2と3が3バッファモードで動作しています。

- EtherCAT masterがSyncManager (SM) 3、バッファ 3からEtherCATフレームによってデータをフェッチする場合 (A)、この読み取り処理が完了すると (B)、EL6695は直ちに他方から次の空きバッファ (ここではバッファ1) への新規データのコピーを開始します。この処理で扱う「Copy Time」は、オンラインで読み取れません。

「自分 (My)」または「リモート (remote)」のどちらが「自分」側になるかは、現在のオンラインCoEがプラ

イマリのものかセカンダリのものかによって変わります。
 - EtherCAT Aが取得するまで、データはそこにとどまります。
 - サイクルタイムが長い場合は、ほぼ1サイクルにわたって比較的古いデータがそこにとどまることがあります。転送のタイミングは、CoE設定によって変更できます。この遅延によって、システムはSM2のEtherCAT Bによって保存されたデータが直後にSM3のA側に内部的に転送されるように最適化を行うことが可能になります。この場合、渡されているEtherCATフレームAがフェッチするデータは、比較的新しいものとなります。
 A側とB側のシステムジッタを最小限にとどめることが重要です。

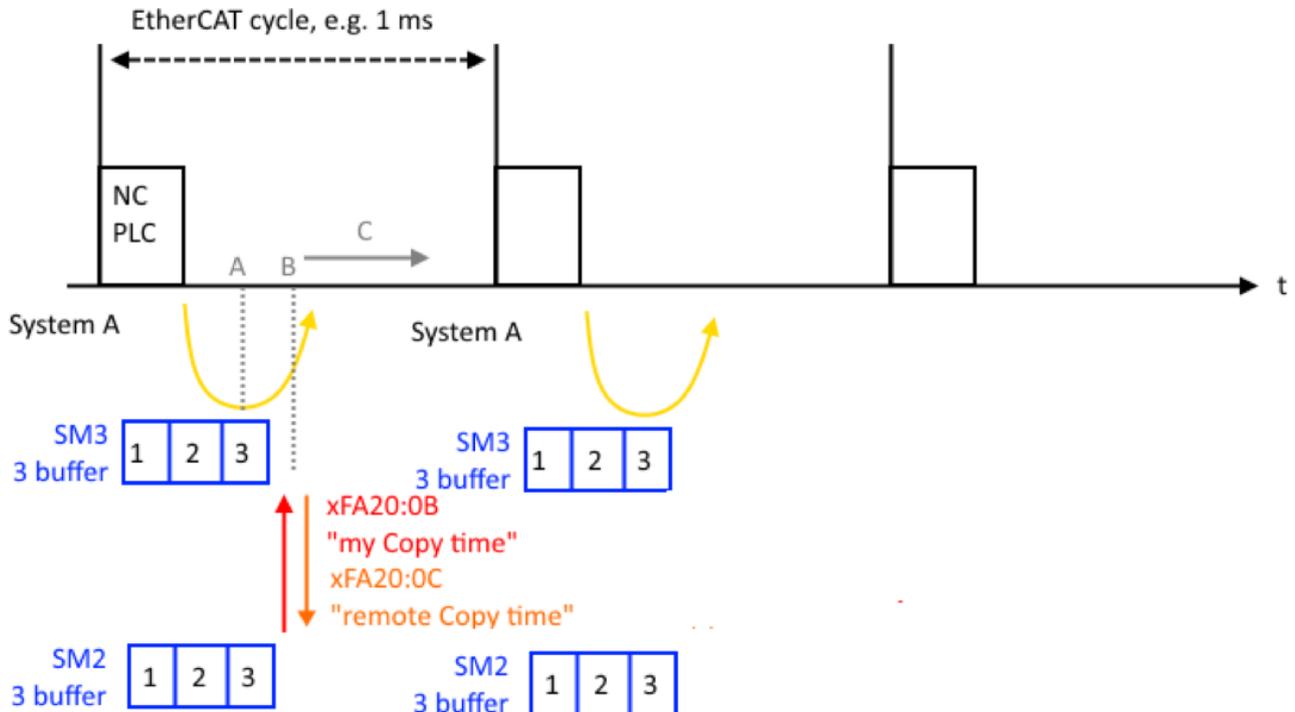


図 132: EL6695のデータ転送シーケンス

- 以下の例は、次のコンフィグレーションでの計測結果を示しています。
- プライマリのPLCがPDOセットを送信
 - EL6695がセットを他方に転送
 - プライマリのPLCがデータをフェッチし、データ形式を変更して再送信
 - EL6695がデータを他方に転送
 - プライマリのPLCがデータをフェッチし、ここまでのPLCサイクルをカウント

この例では、最適化 (PDO遅延、DC同期) は行っていません。

PDOバイト数	タスクサイクルタイム	両方向のタスクサイクル数 (平均値)	かかった転送時間(1方向)	EL6695内での入力および出力データのコピー時間、[Device]ダイアログのCoEオブジェクト0xFA20から (平均値)
200	50 μs	4.4	141.1 μs	14.3 μs
	100 μs	3.03	151.5 μs	14.3 μs
1400	150 μs	8.9	667.5 μs	42.9 μs
	200 μs	6.0	600 μs	42.3 μs

ハード結合したサイクリックデータの転送を実現するためには、両側のEtherCATをディストリビュートクロック結合することで、データ転送中のビート効果を防止することを推奨します。

注意: 実際には、EL6695の2つのフィールドバスには必ずそれぞれのサイクルタイムで動作します。PC A → PC Bの転送時間をパブリッシャ/サブスライバとの直接RealtimeEthernetリンクの転送時間と同程度にするには、タイミングを最適化 (DC同期、シフト時間の調整) する必要があります。この場合でも、最適化は1方向に対してしか行えず、内部転送時間による遅延も発生します。

6.4.2 対称PDO Mapping

一般的なプロセスデータの作成/単純なデータ交換(対称PDO Mapping)

方法:

EL6695ディレクトリツリー内の[IO Inputs]または[IO Outputs]を右クリックして、コンテキストメニューを表示します。[Insert Variable...] (TwinCAT 2)または[Add New Item] (TwinCAT 3)を選択して、変数/プロセスデータを新規作成します(下図を参照)。

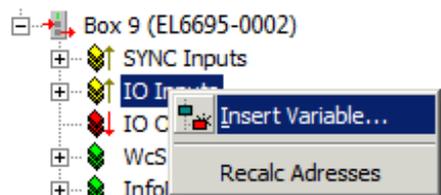


図 133: 「IO inputs」のコンテキストメニュー: 新しい変数の追加

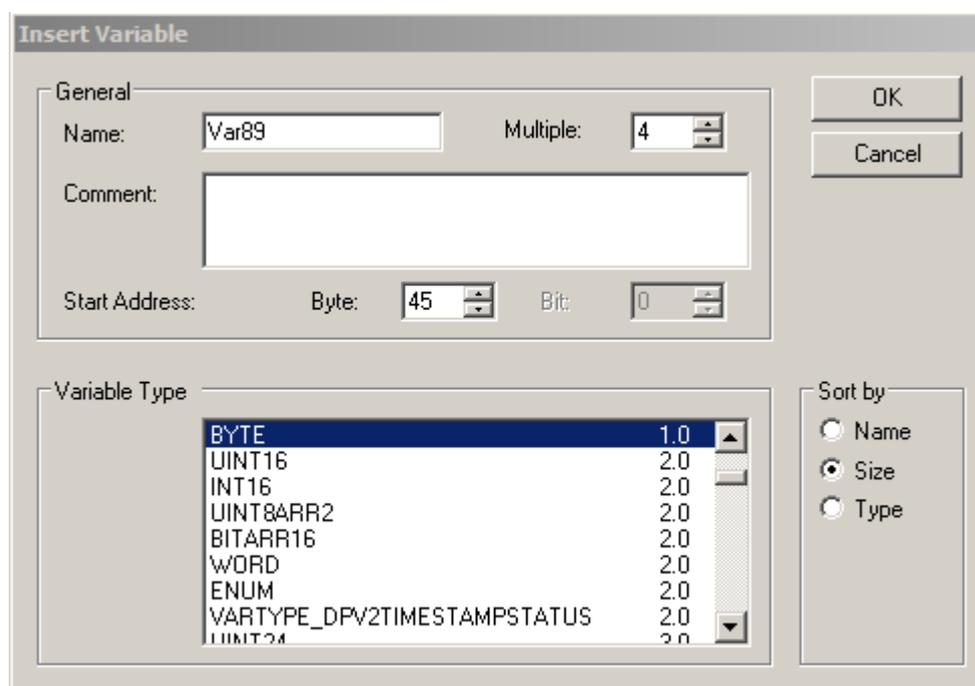


図 134: ダイアログで名前がVar89から始まる新しい変数(ここでは4つのBYTE)を追加

このダイアログでは、変数に名前を割り当て、幅広い種類のデータ型から選択できます。[Multiple]選択ボックスを使用してデータ型が同一の複数の変数を同時に宣言し、定義された開始アドレスを設定できます。上部のダイアログを確定すると、下部にプロセスイメージが表示されます。

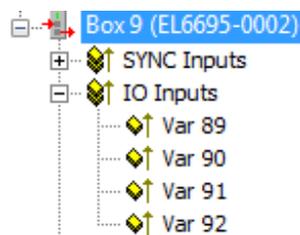


図 135: EL6695のセカンダリに新しく追加された変数

図のように、変数名は自動的にインクリメントされます。出力変数も同様に作成されます。

EL6695のプライマリ(「リモート側」)に、データ交換用の型が同一の4つの適切な出力変数が作成されます。これらの変数名は同一である必要はなく、ここではVar85~Var88となっています。

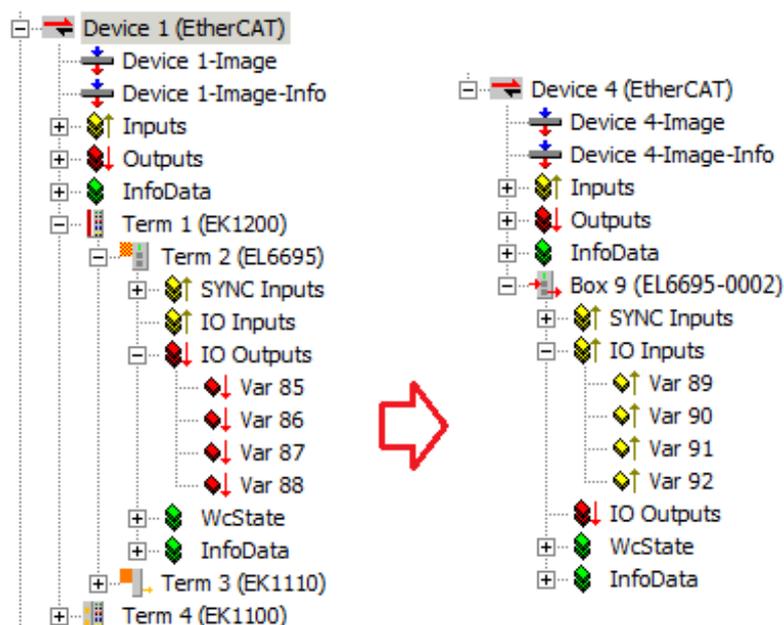


図 136: プロセスデータの作成

ここでも、EL6695のプライマリの出力に4つのBYTE変数を追加します。PLCプログラムなどによって出力変数に対して書き込みを行うと、全ての値は4つの入力変数に連続的に出力されます。例えば、値を出力変数Var86に書き込むと、ターミナルがOP状態であれば、EL6695のリモート側のマッピング内で対応する入力変数Var90に同一の値が書き込まれます。

2つのコントロールPLC1からPLC2への転送方向

通常、ターミナルは2つのコントローラ間のデータ転送に使用されます。ブリッジが出力を入力として、入力を入力としてマッピングすると、結果として以下のような転送が行われます。

- ・ 出力(PLC1) → 出力EL6695 ⇒ 入力EL6695-0002リモート側 → 入力(PLC2)
- ・ 入力(PLC1) ← 入力EL6695 ⇒ 出力EL6695-0002リモート側 ← 出力(PLC2)

それぞれの場合において、EL6695ブリッジターミナルはバイト順に出力値を入力値に、入力値を出力値にマッピングします(他のコンフィグレーションや設定は不要)。このため、使用する変数のデータ型は、それぞれの側で同一である必要があります。

コンフィグレーションインターフェイス/拡張タブによる自動コンフィグレーション:

EL6695拡張タブの[Create Configuration]ボタンによって、作成された変数が他方からオンラインで読み取れるようになるため、他方で個別に作成する必要がありません。プロセスイメージが大きい場合は、この機能を使用することをお勧めします。この機能は、プラグイン[EL6695]がないと使用できません。さらに、EL6695の動作できる状態であり、両方から「オンライン」でアクセスする必要があります。TwinCAT 3については、プライマリに変数を作成し、セカンダリに自動的にミラーリングする手順を以下に記載します。

- ・ 前提条件: TwinCATが「FreeRun」/「Config Mode」であり(右下のアイコン 、 で確認)、[Device] (Term)にEL6695のプライマリが存在していること。
- ・ A) プライマリで変数を作成します(10の入力変数および12の出力変数など)。

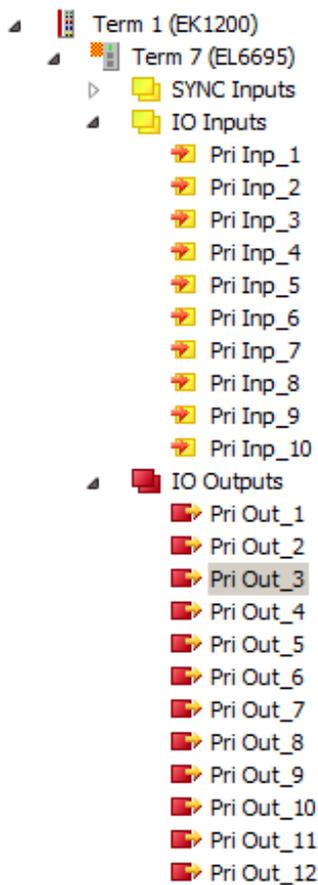


図 137: EL6695のプライマリに適用した変数の例

- ・ B) プライマリで「Terminal」を選択します。[EL6695]タブの[Process Data]にある[Create configuration]ボタンを押すと、他方で変数の「ミラーリング」されたセットが作成されます。これらはターミナルによってセカンダリに内部的に作成されます。このため、他方の変数はまだ表示されません。

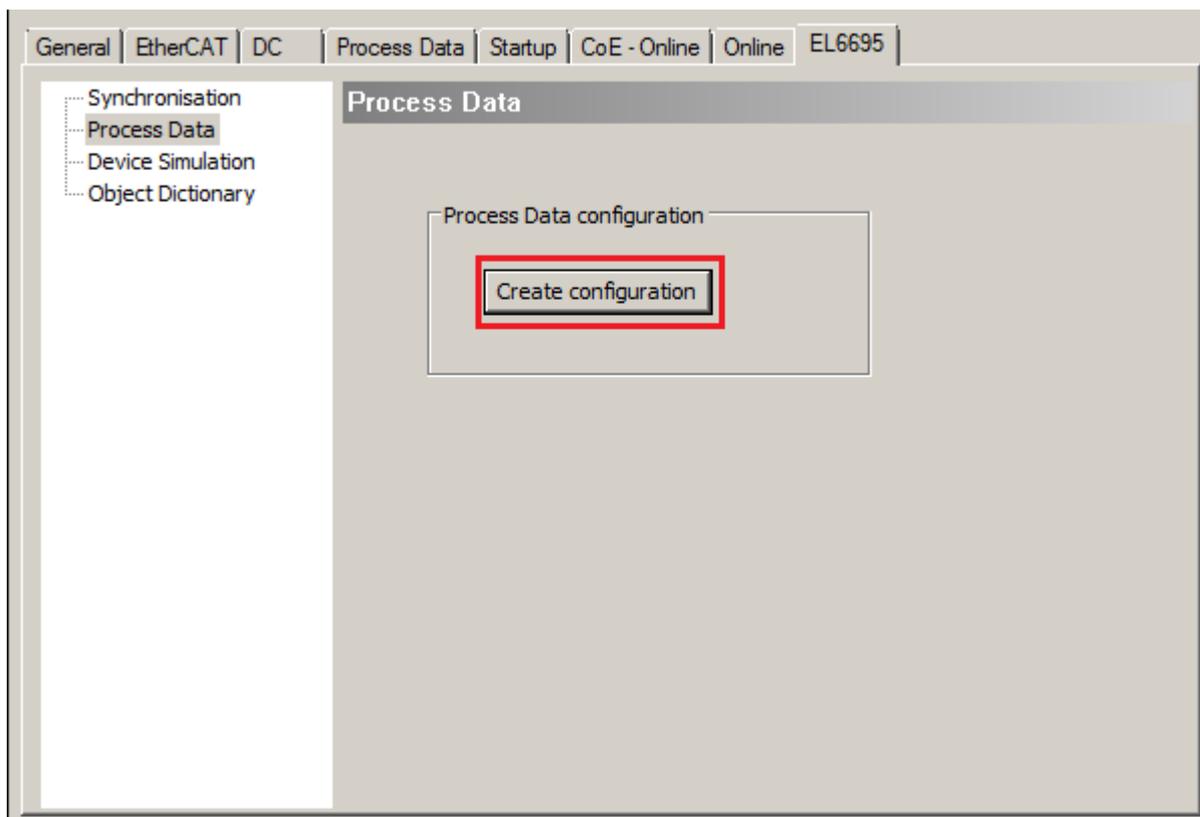
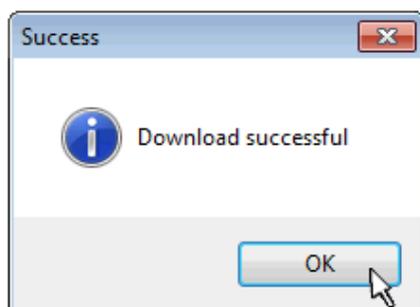
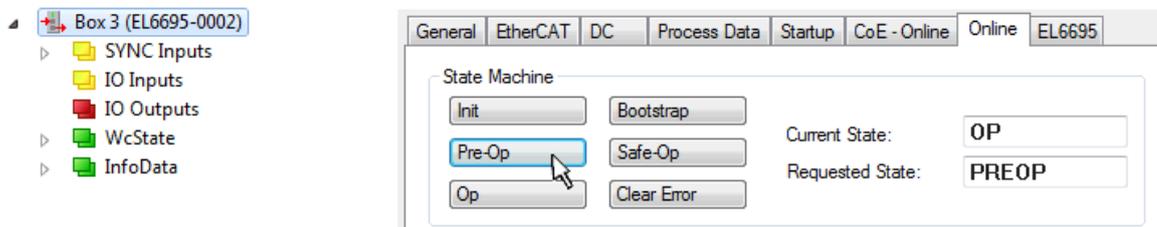


図 138: [EL6695] タブ内の [Process Data] の [Create configuration]

- ・ C) この操作が正常に実行されると、メッセージが表示されます。



- ・ D) ここで、変数のデータが内部的な初期化によって引き継がれるようにするために、ターミナルはセカンダリのデバイスに対して新たにスキャンを行う必要があります。
- ・ E) セカンダリでは [Terminal] が選択され、PREOP オンライン状態である必要があります。そうでない場合は、[Online] タブの [Pre-Op] をクリックして PREOP 状態にする必要があります。



- ・ F) [Process Data] タブの [Load PDO info from device] をクリックすると、他方から「ミラーリング」された変数設定が作成され、Solution Explorer 内に表示されるようになります(左)。

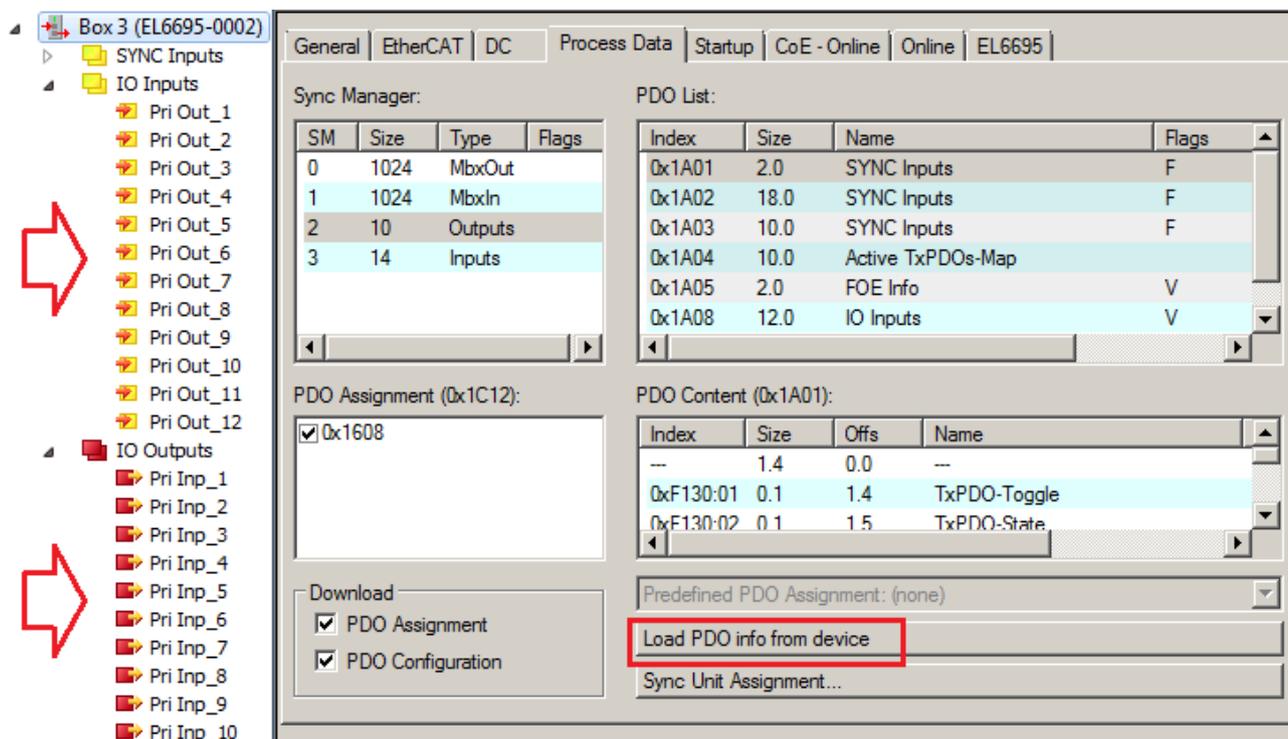


図 139: [Load PDO info from device]でプライマリの変数をセカンダリにミラーリング転送

- ・ G) 必要に応じて、TwinCATメニューの[Reload Devices] ()を使用してコンフィグレーションを新規にロードします。

I/O変数のセット(PDOコンフィグレーション)がセカンダリに存在している場合、ここで説明した手順にしたがって変数をプライマリに作成することも可能です。

● スキャンができない/スキャンが適切ではない場合

i セカンダリに変数を作成するには、[Create configuration]のステップB)および and C)の後でプライマリのターミナルを手動でINIT状態に設定してから、再度OP状態に設定する必要があります ([Online]タブ: 「Init」をクリックしてから「Pre-Op」、「Safe-Op」、「Op」をクリック)。これにより、セカンダリでEL6695 (-0002)の各変数を挿入できます。これ以外の手順は、前述のE)以降と同様です。

● サンプルプログラムの使用

i 本取扱説明書には、当社製品の特定の用途領域向けのサンプルアプリケーションが含まれています。ここで提供されているアプリケーションノートは、当社製品の一般的な特徴に基づいたものであり、サンプルとしてのみお使いいただけます。本取扱説明書に含まれているアプリケーションノートは、決して特定のアプリケーションを対象としたものではありません。このため、お客様の責任の下、特定のアプリケーションに対して製品が適しているかどうかを評価および判断してください。当社は、本取扱説明書に含まれているソースコードの完全性および正確性に対して、いかなる責任も負いません。当社は、本取扱説明書の内容を随時変更する権利を有する一方で、誤りおよび情報の不足については一切の責任を負いません。

→ コンフィグレーション例: <https://infosys.beckhoff.com/content/1033/el6695/Resources/zip/3521535883.zip>

6.4.3 選択的PDO Mapping

選択的PDO Mapping

選択的PDO Mappingにより、ユーザはカスタマイズされたサイクリックデータ交換を設定できます。これを行うために、プロセスデータの「完全な」セットがターミナルの一方に作成されます。他方には、周期的な読み取りのために、このセットのサブセットを定義できます。この処理は、選択的PDO Mappingと呼ばれます。

デフォルトのPDO割り当ておよびマッピングは削除され、ユーザがカスタムのPDO割り当ておよびマッピングを作成する必要があります。これを行うためには、PDO AssignまたはPDO Mappingの構造を知ることが重要です。

基本的には(標準では)、2つのCoEオブジェクト0x1C12「Rx PDO Assign」と0x1C13「Tx PDO Assign」があり、ここでセカンダリおよびプライマリのデータ入力(RxPDO)とデータ出力(TxPDO)の割り当てを定義します。これらのCoEオブジェクトの内容は、入出力のマッピングが定義されるCoEオブジェクト0x1608「IO RxPDO Map」と0x1A08「IO TxPDO Map」への参照です。これら4つのCoEオブジェクト0x1C12/0x1C13および0x1608/0x1A08には、RWフラグ(RW = 読み取り/書き込み)が付けられています。2つのMappingオブジェクト0x1608および0x1A08は、初期状態ではプロセスデータ(構造体)が格納されているCoEオブジェクト0x7000「PD Outputs」および0x6000「PD Inputs」(プロセスデータ入力/出力)を参照します。「初期状態」では、ユーザがプライマリまたはセカンダリに新しい入力または出力データを追加する際に、TwinCATによる自動構成を参照します。これは、[Startup]タブでも確認できます。ここで、PREOPからSAFEOP (P → S)への遷移における、オブジェクト参照のリンクがすべて作成されます。デバイス内において、OP状態で入力および出力データのコピー処理が実行されるため、2つの「最終的な」CoEオブジェクト0x6000および0x7000は「RO」(読み取り専用)となります。

Index	Name	Flags	Value
1000	Device type	RO	0x00001389 (5001)
1008	Device name	RO	EL6695 <SECONDARY>
1009	Hardware version	RO	00
100A	Software version	RO	02
1018:0	Identity	RO	> 4 <
10F4:0	External synchronization status	RO	> 18 <
1608:0	IO RxPDO-Map 8	RW	> 1 <
1608:01	SubIndex 001	RW	0x7000:01, 8
1801:0	TxPDO-Par External Sync Compact	RO	> 6 <
1802:0	TxPDO-Par External Sync	RO	> 6 <
1803:0	TxPDO-Par External Sync(32 Bit)	RO	> 6 <
1A01:0	TxPDO-Map External Sync Compact	RO	> 5 <
1A02:0	TxPDO-Map External Sync	RO	> 8 <
1A03:0	TxPDO-Map External Sync(32 Bit)	RO	> 8 <
1A04:0	Active TxPDOs-Map	RW	> 5 <
1A05:0	FOE Info-Map	RW	> 1 <
1C00:0	Sync manager type	RO	> 4 <
1C12:0	RxPDO assign	RW	> 1 <
1C12:01	SubIndex 001	RW	0x1608 (5640)
1C13:0	TxPDO assign	RW	> 2 <
1C32:0	SM output parameter	RO	> 32 <
1C33:0	SM input parameter	RO	> 32 <
7000:0	PD Outputs	RO	> 1 <
7000:01	SubIndex 001	RO P	0x00 (0)
F000:0	Modular device profile	RO	> 2 <

図 140: 0x1C12による1バイト変数の0x7000「PD Outputs」への基本的なマッピング

(初期状態の)リンクされたオブジェクトの概要:

- ・ RxPDO Assign: 0x1C12 → RxPDO Map 0x1608 → 0x7000 PD Outputs
- ・ TxPDO Assign: 0x1C13 → TxPDO Map 0x1A08 → 0x6000 PD Inputs

● CoEオブジェクトのマッピング

I 0x1608は0x7000をポイントし、0x1A08は0x6000をポイントします。オブジェクト0x7000および0x6000は、それぞれターミナルのブリッジ先にミラーリングされます。

例えば、オブジェクト0x7000:0E PD Outputsがプライマリに存在する場合、オブジェクト0x6000:0E PD Inputsはデータ交換用にセカンダリ側に作成する必要があります。

TwinCATでの手順

選択的マッピングを作成するには、最初にターミナルの[Process Data]タブのOutputs, InputsおよびDownloadのすべてのオプションのチェックマークを外して、既存のPDOコンフィグレーションとその割り当てを無効にする必要があります。

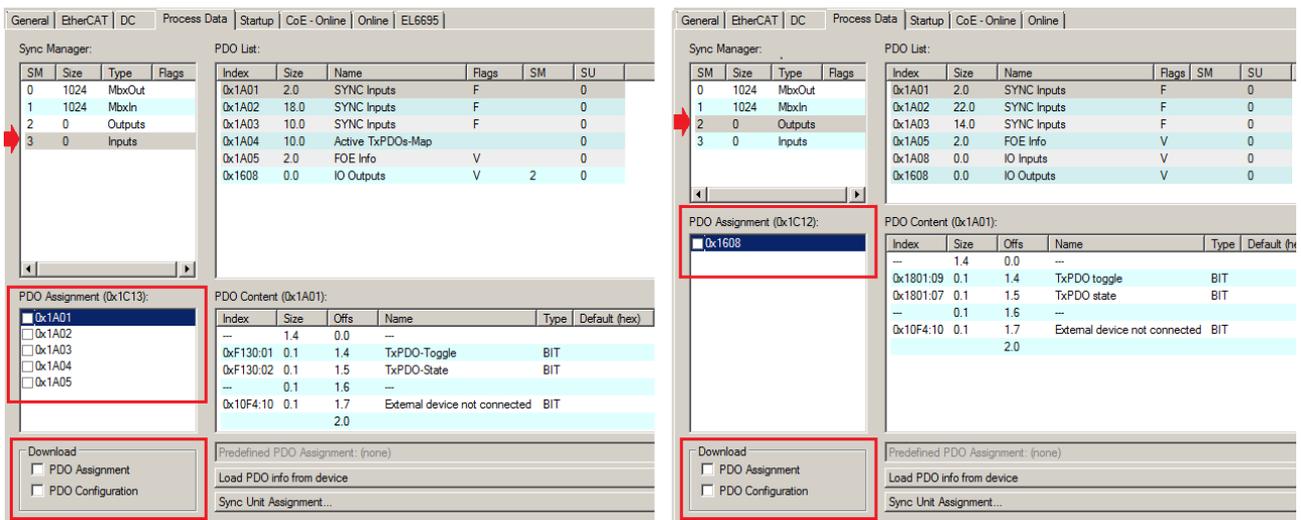


図 141: 選択的マッピングでのプロセスデータ

これにより、[Startup]タブ内の既存のエントリも削除され、ターミナルにはプロセスイメージがなくなります。この状態で、プロセスイメージを以下の2つの手順で再定義します。

ターミナルの[Startup]タブ内で、エントリを新規作成する必要があります。このタブの右下にある[New...]を選択します。

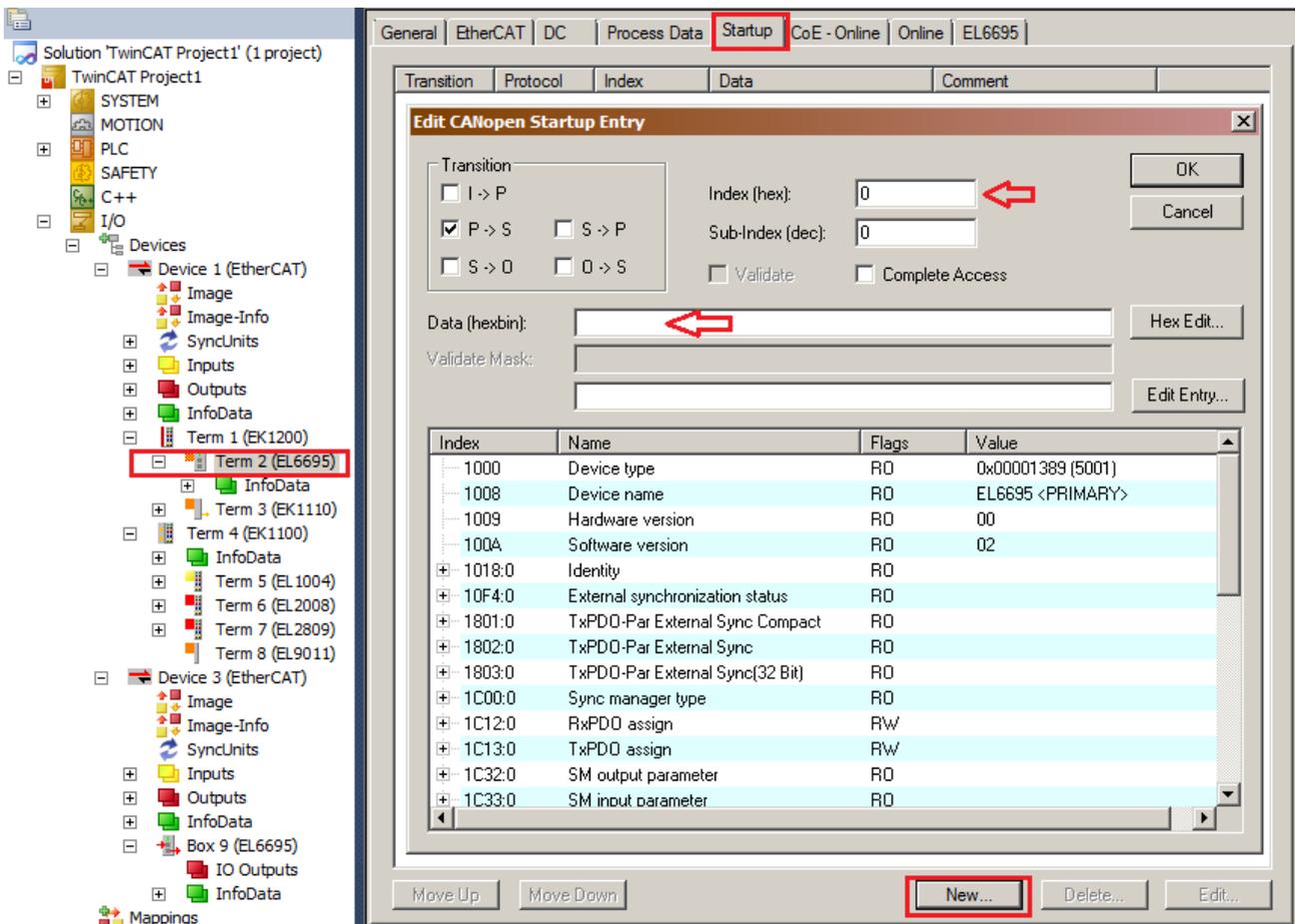


図 142: 出力変数の新規作成(この例ではEL6695のプライマリ)

例として、サイズが8、16、32、そして再度8の合計4つの出力変数を作成します(推奨シーケンス)。

これを行うために、この例では次の手順で0x1614オブジェクト(0x1600~0x161Fの範囲から選択可能)を作成します([Complete Access]にチェックマークを入れ、コメントに「IO RxPDO Map」などを入力)。

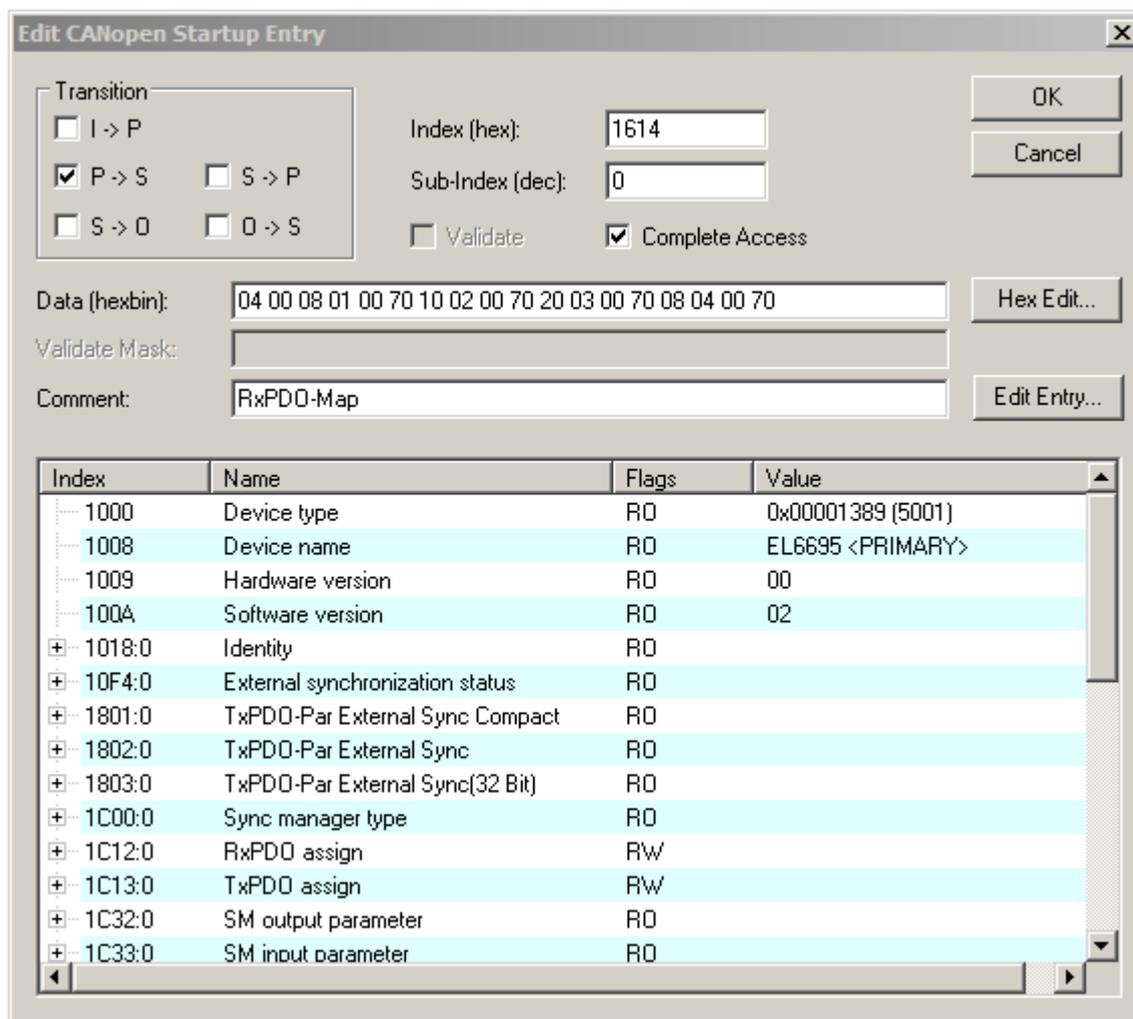


図 143: 0x1614オブジェクトのセットアップ

・ 入力: 04 00 08 01 00 70 10 02 00 70 20 03 00 70 08 04 00 70

入力データの意味:

04 00	サブインデックス番号
08 01 00 70	オブジェクト0x7000、インデックス1、長さ8
10 02 00 70	オブジェクト0x7000、インデックス2、長さ16
20 03 00 70	オブジェクト0x7000、インデックス3、長さ32
08 04 00 70	オブジェクト0x7000、インデックス4、長さ8

これにより、プログラムオブジェクトや構造体などのデータ構造を表現できます。構造体変数をEL6695のPDOにリンクするには、先にTC3でグローバルデータ型に変換する必要があります。

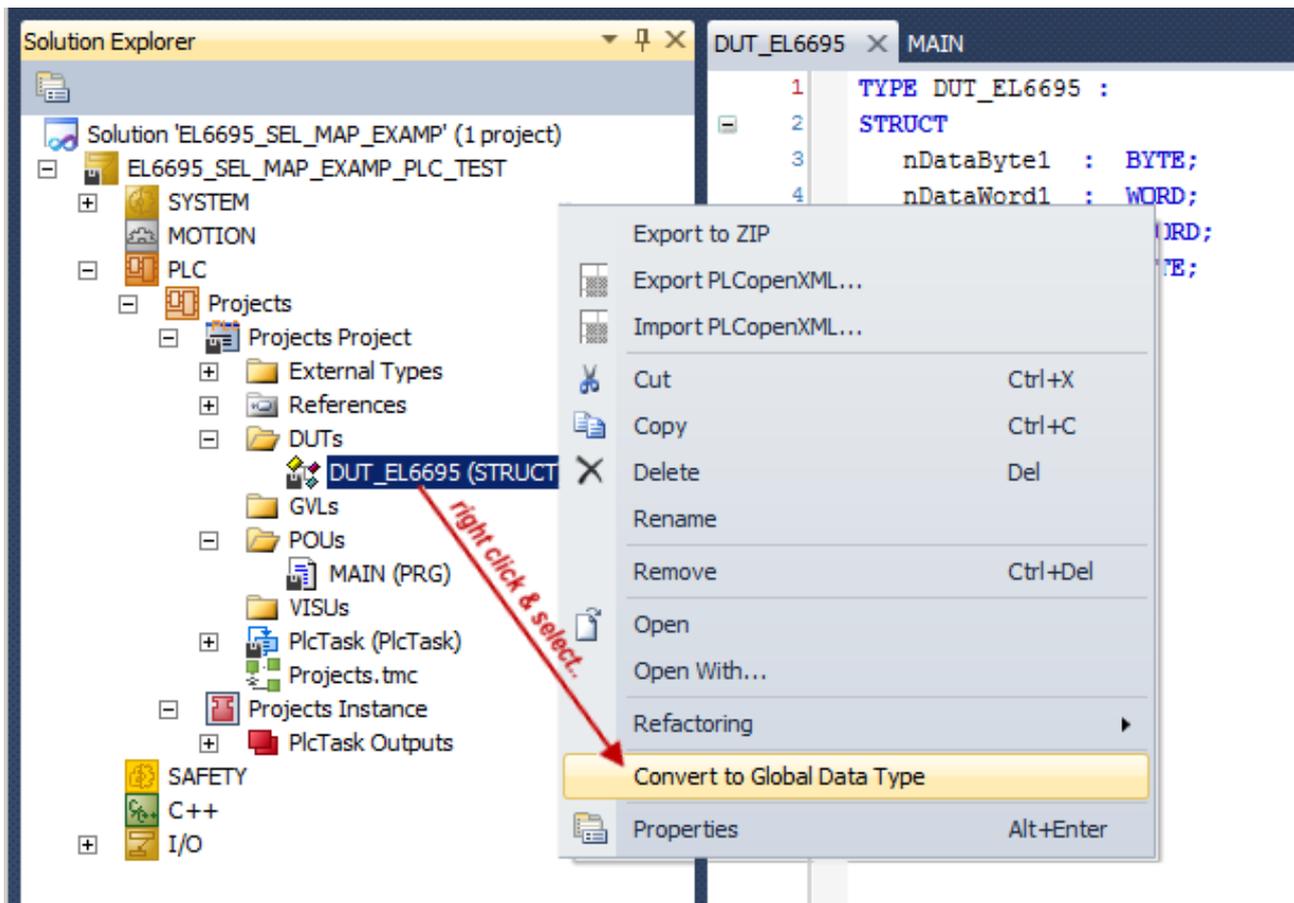


図 144: グローバルデータ型への変換

2番目のステップ [Edit CANopen Startup Entry] ダイアログを使用して、以下の入力を行います。

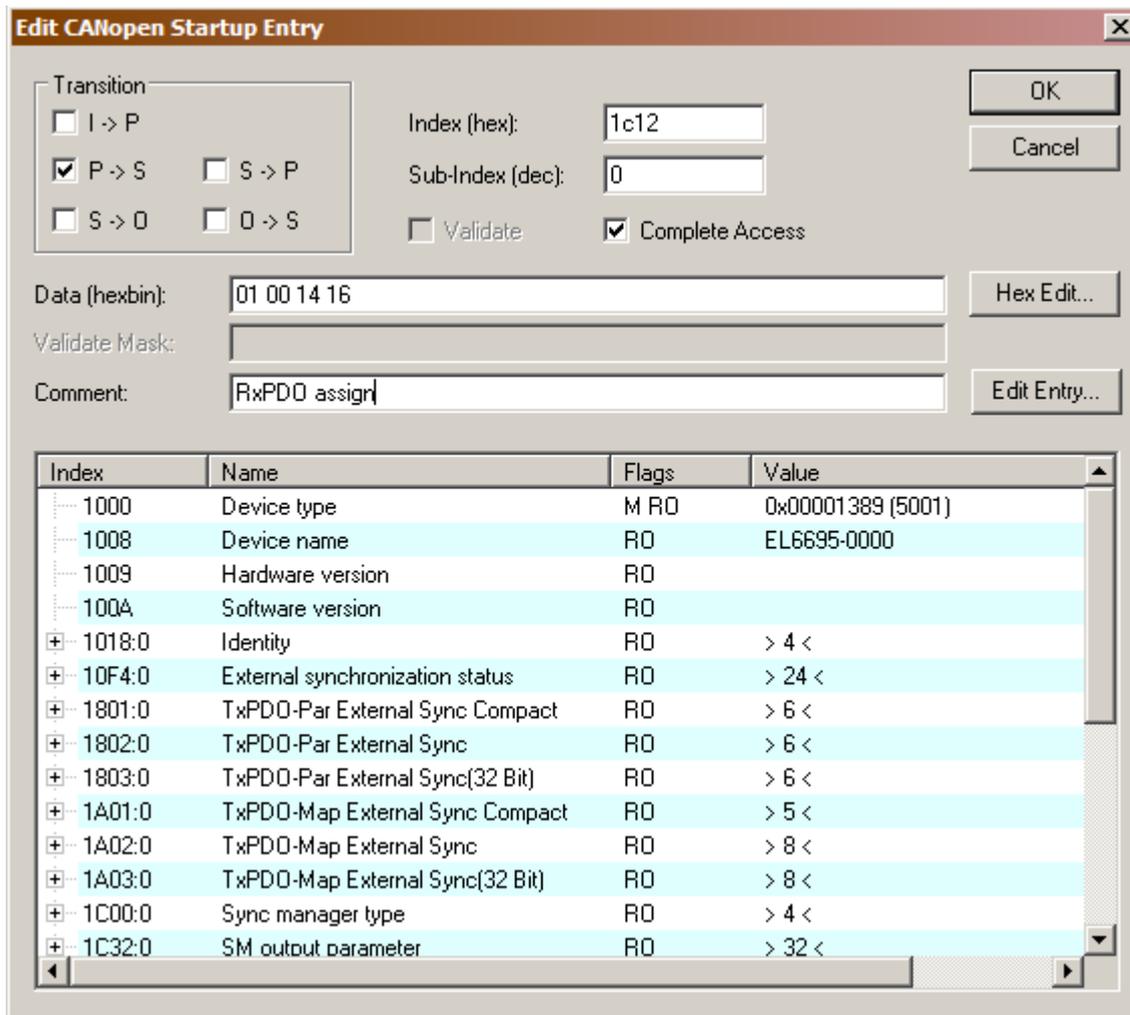


図 145: RxPDO用のオブジェクト0x1C12の作成

このため、作成するCoEスタートアップエントリは、インデックス0x1C12、サブインデックス0（オブジェクトの最初のサブインデックス）となります。[Complete Access]にチェックマークを入れ、入力値がデータ型WORD、DWORDなどとして解釈されないようにします。PREOPからSAFEOPへの遷移中に、割り当てを行う必要があります。チェックマークは「P → S」の遷移のみをセットします。[Comment]には、説明のコメントを追加でき、これがオブジェクト名としてCoE概要に表示されます。CoEオブジェクトの内容は、[Data (16進数)]で記述します。

最初の2つのブロックは、オブジェクトのサブインデックス数を指定します(Hi/Loバイト切り替え: 01 00 → サブインデックス数 = 1)。この後に、最初のサブインデックス01とその内容14 16が続きます → 0x1614。

・ 入力: 01 00 14 16

これは、前述の参照の1つです。ここでは、作成済みの「RxPDO Map」オブジェクト(0x1614)を参照していません。

ここで、[Startup]タブは次のように表示されます。

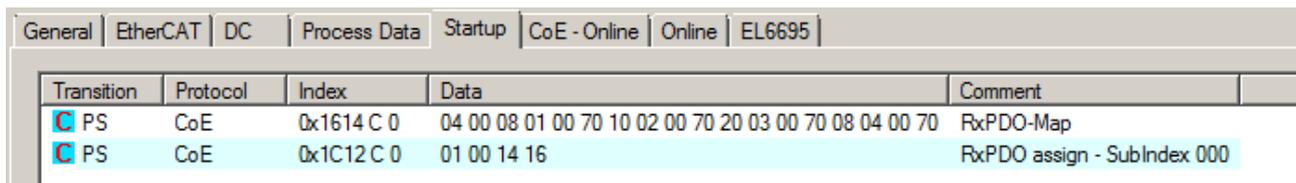


図 146: プライマリの新規PDO Mappingおよび割り当てオブジェクト

オブジェクト0x1C12に対する「RxPDO assign」が、必ずRxPDO Mapエントリリストの後となることが重要です。これには、必要に応じて複数の参照を含めることができます。これにより、データセットは次のようになります: 06 00 03 16 1A 16 1E 16 08 16 04 16 .. など。

以下の手順で、コンフィグレーション内で作成された変数を表示します。

1. メニュー[TwinCAT] → [Reload Devices]を選択
2. ターミナルの[Process Data]タブ内の[Load PDO info from device]
3. [Update List]でCoEオンラインオブジェクトを表示

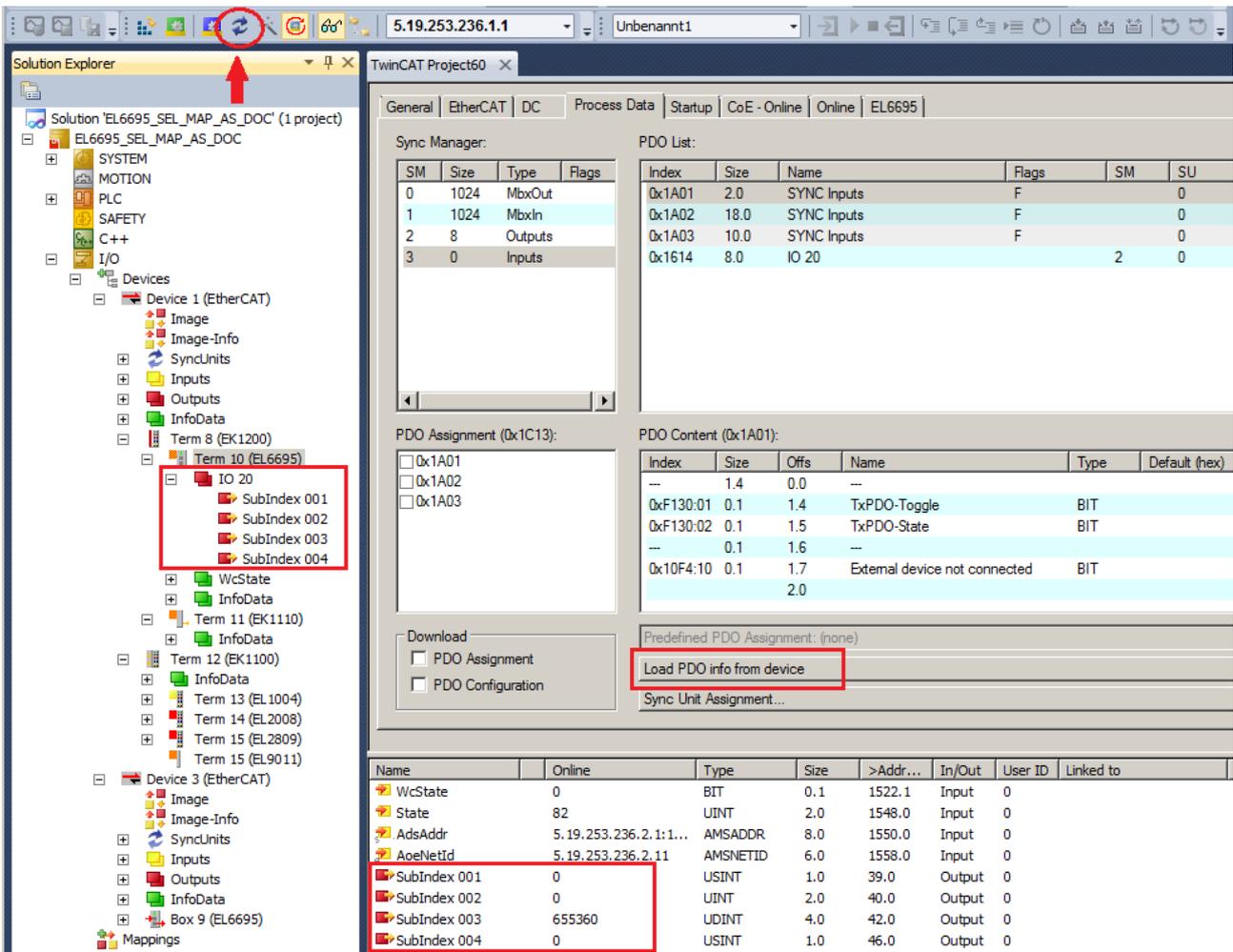


図 147: 新規作成されたプロセスデータ:オブジェクト0x1608

Index	Name	Flags	Value
+ 1803:0	TxPDO-Par External Sync(32 Bit)	RO	> 6 <
+ 1C00:0	Sync manager type	RO	> 4 <
- 1C12:0	RxPDO assign	RW	> 1 <
1C12:01	SubIndex 001	RW	0x1614 (5652)
+ 1C13:0	TxPDO assign	RW	> 0 <
+ 1C32:0	SM output parameter	RO	> 32 <
+ 1C33:0	SM input parameter	RO	> 32 <
+ F000:0	Modular device profile	RO	> 2 <
F008	Code word	RW	0x00000000 (0)
+ F010:0	Module list	RO	> 1 <
+ F130:0	TxPDO-Parameter	RO	> 2 <
+ F630:0	Active TxPdo Info	RO	> 4 <
+ F640:0	Remote Write Cycles	RO	> 3 <
+ F650:0	FOE Info	RO	> 2 <
+ F800:0	Device Config	RO	> 3 <
+ F820:0	ADS Server Settings	RO	> 2 <
+ F821:0	EL6695 ADS Settings	RO	> 2 <
+ FA20:0	Device Diag	RO	> 29 <
- 1614:0	IO RxPDO-Map 20	RW	> 4 <
1614:01	SubIndex 001	RW	0x7000:01, 8
1614:02	SubIndex 002	RW	0x7000:02, 16
1614:03	SubIndex 003	RW	0x7000:03, 32
1614:04	SubIndex 004	RW	0x7000:04, 8
+ 1A01:0	TxPDO-Map External Sync Compact	RO	> 5 <
+ 1A02:0	TxPDO-Map External Sync	RO	> 8 <
+ 1A03:0	TxPDO-Map External Sync(32 Bit)	RO	> 8 <
+ 1A04:0	Active TxPDOs-Map	RW	> 5 <
+ 1A05:0	FOE Info-Map	RW	> 1 <
- 7000:0	PD Outputs	RO	> 4 <
7000:01	SubIndex 001	RO P	0x00 (0)
7000:02	SubIndex 002	RO P	0x0000 (0)
7000:03	SubIndex 003	RO P	0x00000000 (0)
7000:04	SubIndex 004	RO P	0x00 (0)

図 148: 選択的に作成された変数および対応する参照の表示[CoE Online]

選択的PDO変数の作成

EL6695ブリッジターミナルのブリッジ先(ここではセカンダリ)で、選択した「変数」を必要に応じて作成できます。これを行うには、Solution Explorer内で必要な変数を追加します。

この例ではプライマリに1-2-4-1バイトの構造体があり、図のように、そのうちの2バイトおよび4バイトの変数「WORD」型と「DWORD」型のみがプロセスデータ全体のサブセットとして提供されます。

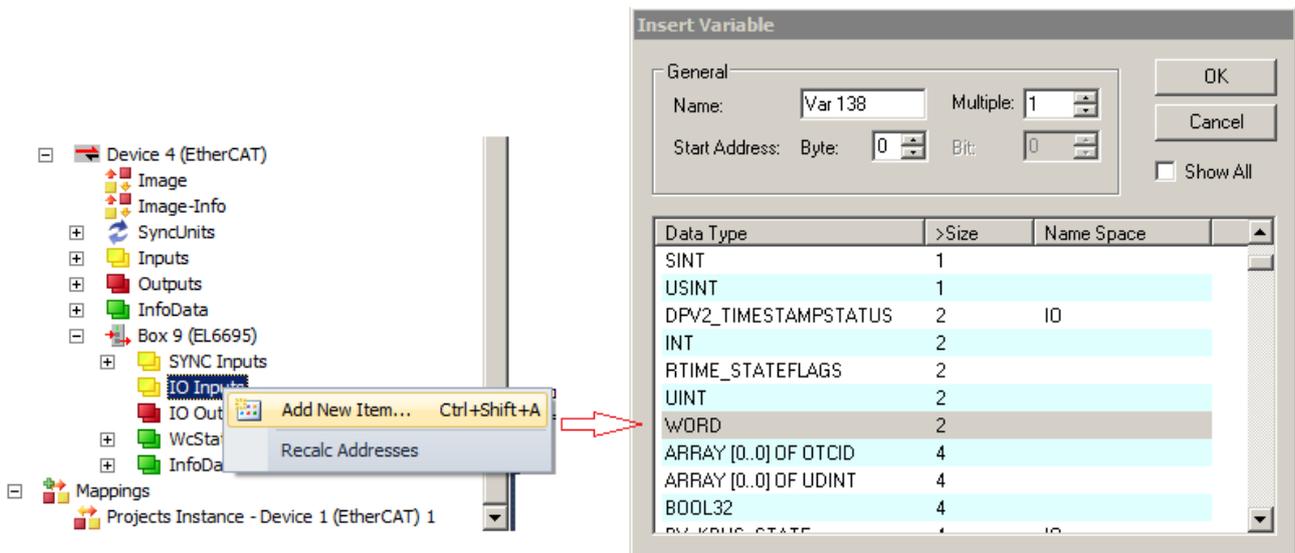


図 149: 選択的PDO変数の追加

図で示すように、DWORD変数の手順も同様です。

以下のようにスタートアップエントリを完了し、WORD型のサブインデックス値02、およびDWORD型のサブインデックス値03を追加します。

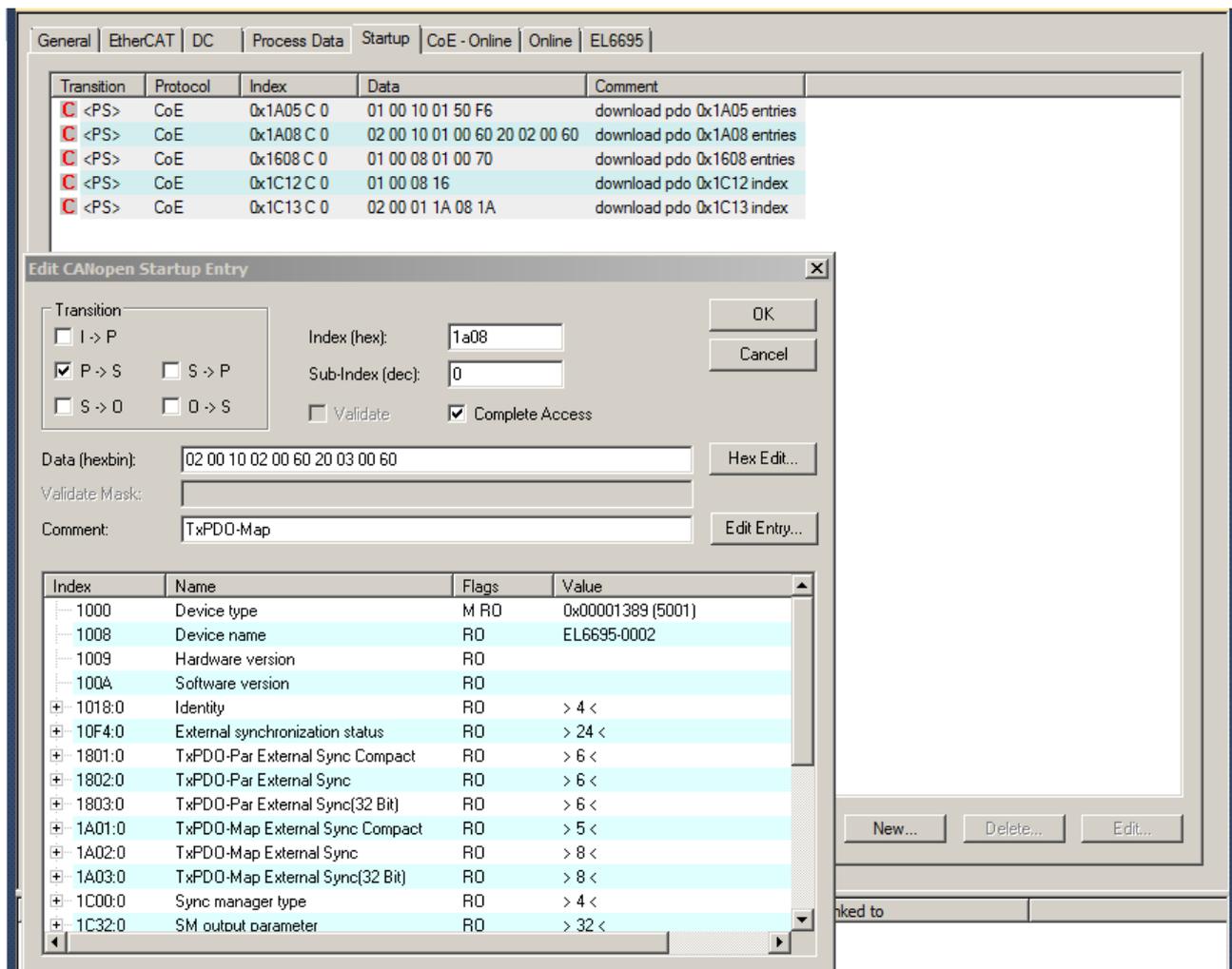


図 150: 上書き用のスタートアップエントリ0x1A08(参照インデックスを修正01 → 02、02 → 03)

- ・ 入力: 02 00 10 02 00 60 20 03 00 60

入力データの意味:

02 00	サブインデックス番号
10 02 00 60	オブジェクト0x6000、インデックス2、長さ16
20 03 00 60	オブジェクト0x6000、インデックス3、長さ32

「Reload」命令をさらに実行し、コンフィグレーションを有効にすると、PLCプログラムで選択的PDO Mappingを使用できます。ターミナル固有のシステム特性により、リアルタイムデータ交換には特殊なバッファリングメカニズムが使用されているため、出力変数を必ず先頭の1バイトと終端の1バイト(いわゆる開始/停止バイト)で設定する必要があります。これらの「フィルバイト」変数に対して、PLC変数へのリンクおよびそれらデータの使用は不要です。

EL6695とその他のEtherCATマスタのアプリケーションごとに、コンフィグレーション手順を実施する必要があります。

開始/停止バイトの使用に関する注意事項

通常、ベッコフは出力データ内での選択的マッピングについて、先頭に開始バイト、終端に停止バイトを作成することを推奨しています。これは、プロセスデータの内部メモリ管理に関して、最後のバイトへのアクセスによってターミナルの次のEtherCATバッファへの内部切り替えが発生し、その後この出力パッケージに対して読み取りアクセスができなくなるという技術的な理由によるものです。これは、出力バッファへの読み取りアクセスが「上から下に」選択的に行われる場合は問題とはなりません。ただし、変数が他のセクションから連続的に読み取られ(ほとんどのアプリケーションが当てはまります)、最後のバイトが読み取られると、バッファがロックされ、その後はデータを取得できなくなります。実際のプロセスデータの後に「空の」バイトを追加することで、バッファの意図しない切り替えを防止します。

下図では、多くの場合で開始/停止バイトが実際には不要であることを示しています。

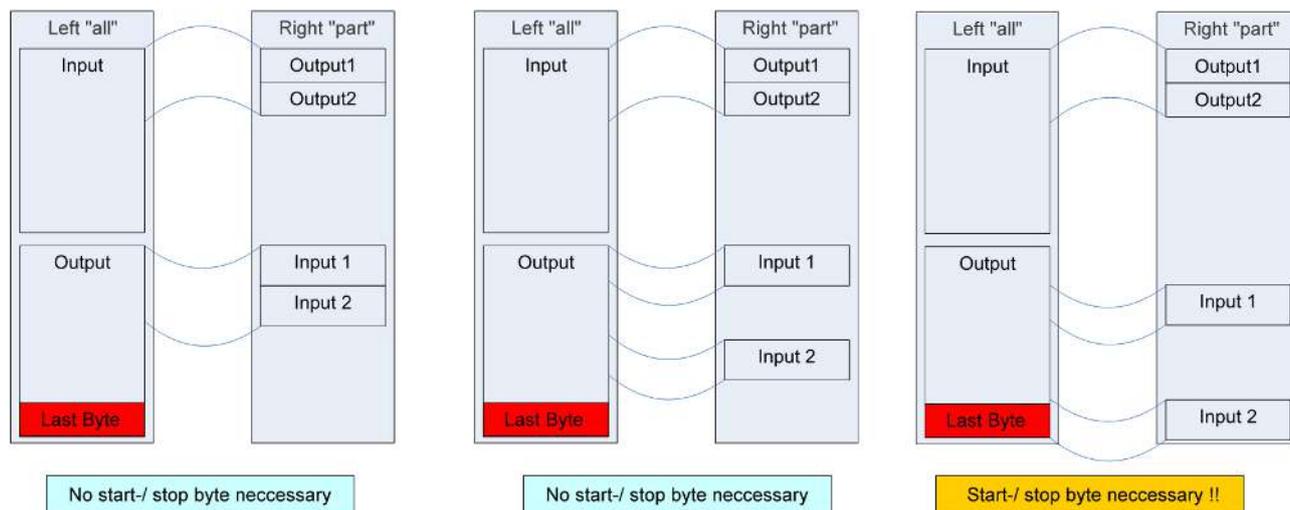


図 151: 開始/停止バイトの使用

● 開始および停止バイトの挿入

i アプリケーションの耐用年数を超えてEL6695を柔軟に活用している際の想定外の動作やコンフィグレーションの変更を避けるため、通常、ベッコフは出力変数の先頭に1バイト、終端に1バイトを設定することを推奨しています。

● サンプルプログラムの使用

i 本取扱説明書には、当社製品の特定の用途領域向けのサンプルアプリケーションが含まれています。ここで提供されているアプリケーションノートは、当社製品の一般的な特徴に基づいたものであり、サンプルとしてのみお使いいただけます。本取扱説明書に含まれているアプリケーションノートは、決して特定のアプリケーションを対象としたものではありません。このため、お客様の責任の下、特定のアプリケーションに対して製品が適しているかどうかを評価および判断してください。当社は、本取扱説明書に含まれているソースコードの完全性および正確性に対して、いかなる責任も負いません。当社は、本取扱説明書の内容を随時変更する権利を有する一方で、誤りおよび情報の不足については一切の責任を負いません。

→ ご要望により、選択的マッピングのサンプルプログラムをご提供します。

<https://infosys.beckhoff.com/content/1033/el6695/Resources/zip/1421824651.zip>

<https://infosys.beckhoff.com/content/1033/el6695/Resources/zip/1421826315.zip>

6.4.4 FSoE転送

周期プロセスデータ通信として行うFSoE転送は、2つの異なる方法で実装可能です。

- ・ セーフティモジュール/スロットを含むESIを作成し、EmulationModeでロード。関連個所を参照してください。
- ・ 「外部デバイス」のセーフティ接続を設定。
- ・ 注意：EL6695はライセンス認証を必要とするセーフティ関連デバイスではなく、単なる転送リンクの要素です。当然のことながら、EL6695の「背後」に隠れている安全デバイスに対してETG FSoEテストを実施する場合は、FSoE ETGテストがEL6695を「セーフティ」デバイスとして認識するようにEL6695のPD0を設定する必要があります。

詳細情報については、関連データ(TwinCATバージョン、必要なパフォーマンス値、サイクルタイム、データ量など)をご用意の上、ベッコフサポートまでお問い合わせください。

6.5 ローカルメールボックスプロトコル

EL6695によってローカルで処理され、転送しない非周期的データ。通常、EtherCATメールボックス経由で送信。

6.5.1 CoE – Can over EtherCAT

CoE – Can over EtherCAT

処理：ローカル、転送なし

PLCプログラム(ST)内では、ファンクションブロックFB_EcCoeSdoReadおよびSDO (service data object) アクセスを使用して、データをEtherCATスレーブのオブジェクトディレクトリから読み込みます。読み込むオブジェクトの選択には、パラメータnSubIndexおよびnIndexを使用します。このブロックは、(標準)ライブラリTc2_EtherCATの一部です(<http://infosys.beckhoff.com>の「TwinCAT 3 Engineering」 → 「PLC」 → 「Libraries」 → 「TwinCAT 3 PLC Lib: Tc2_EtherCAT」 → 「CoE」を参照してください)。

● サンプルプログラムの使用

i 本取扱説明書には、当社製品の特定の用途領域向けのサンプルアプリケーションが含まれています。ここで提供されているアプリケーションノートは、当社製品の一般的な特徴に基づいたものであり、サンプルとしてのみお使いいただけます。本取扱説明書に含まれているアプリケーションノートは、決して特定のアプリケーションを対象としたものではありません。このため、お客様の責任の下、特定のアプリケーションに対して製品が適しているかどうかを評価および判断してください。当社は、本取扱説明書に含まれているソースコードの完全性および正確性に対して、いかなる責任も負いません。当社は、本取扱説明書の内容を随時変更する権利を有する一方で、誤りおよび情報の不足については一切の責任を負いません。

→ ご要望により、CoEのサンプルプログラムをご提供します。

<https://infosys.beckhoff.com/content/1033/el6695/Resources/zip/1421141259.zip>

6.6 送信機能付きのメールボックスプロトコル

6.6.1 CoEアクセス中のAoEの使用例

CoEアクセス中のAoEの使用例

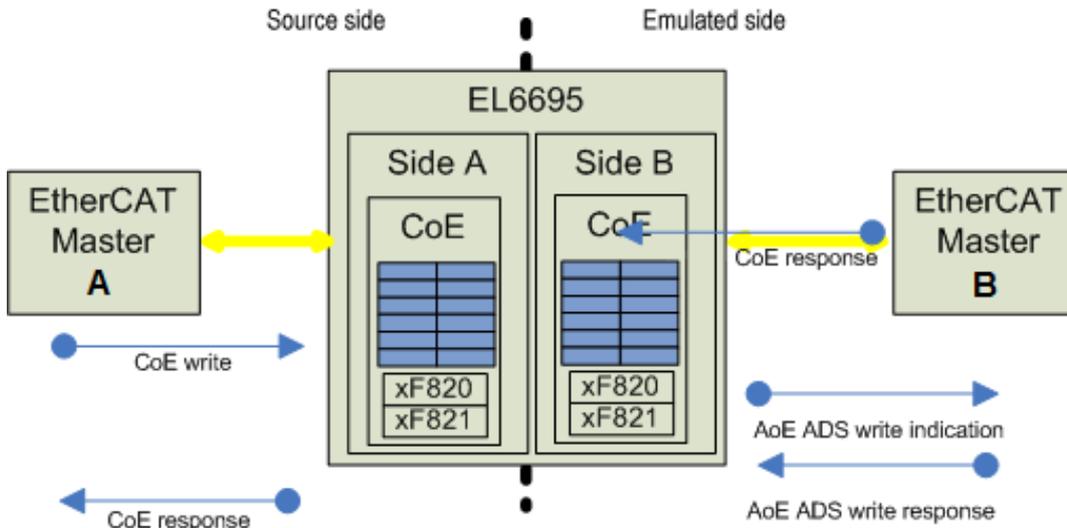


図 152: AoE ADSを使用した操作でのCoE書き込みアクセスを示す図

- EL6695には両側にCoEディレクトリがあり、少なくとも一方は宣言されたユーザ固有のディレクトリです。原則として、このモードでは、EL6695はCoE AとCoE Bの間の接続を確立しません。これにより、2つのコントローラはCoEを使用して非同期的にパラメータを交換できず、一方での書き込みアクセスは他方のマスタに通知できません。このため、以下のデータ交換メカニズムがベッコフADSを使用してEL6695内に実装されています。
 - 仮定: 少なくとも片方がベッコフAoE (ADS over EtherCAT)をサポートするTwinCATシステム、またはEtherCATマスタであるとして。この例では、マスタBがこのタイプのシステムであり、マスタAは単にCoEをサポートしているシステムです。
 - マスタAによるCoE Aへの書き込み/読み取りアクセスが、CoE Bに送信されます。
 - EL6695はCoE B 0xF820からターゲットのAMSNetIDを読み取り、AoE ADS書き込み/読み取り指示を指定されたアドレスに送信します。このアドレスで、サービスがこの指示を取得して処理する必要があります。TwinCAT PLCでのこの指示は、TcSystem.libのFB ADSWRITEIND/ADSREADINDなどです。
 - その後、PLCはADSWRITERES/ADSREADRESを使用してEL6695に応答を送り返します。CoE A 0xF821からのターゲットアドレスとして、自身のAMSNetIDを使用します。スタートアップ中、EtherCATマスタはこのAMSNetIDをAoE Init CmdとしてEL6695にロードする必要があります。
 - これで、EL6695はCoE書き込み/読み取り応答をマスタAに戻せるようになります。
 - この処理では、送信時間を考慮する必要があります。
 - このAoEメカニズムは、AMSNetIDが格納されているCoE領域内で動作します。

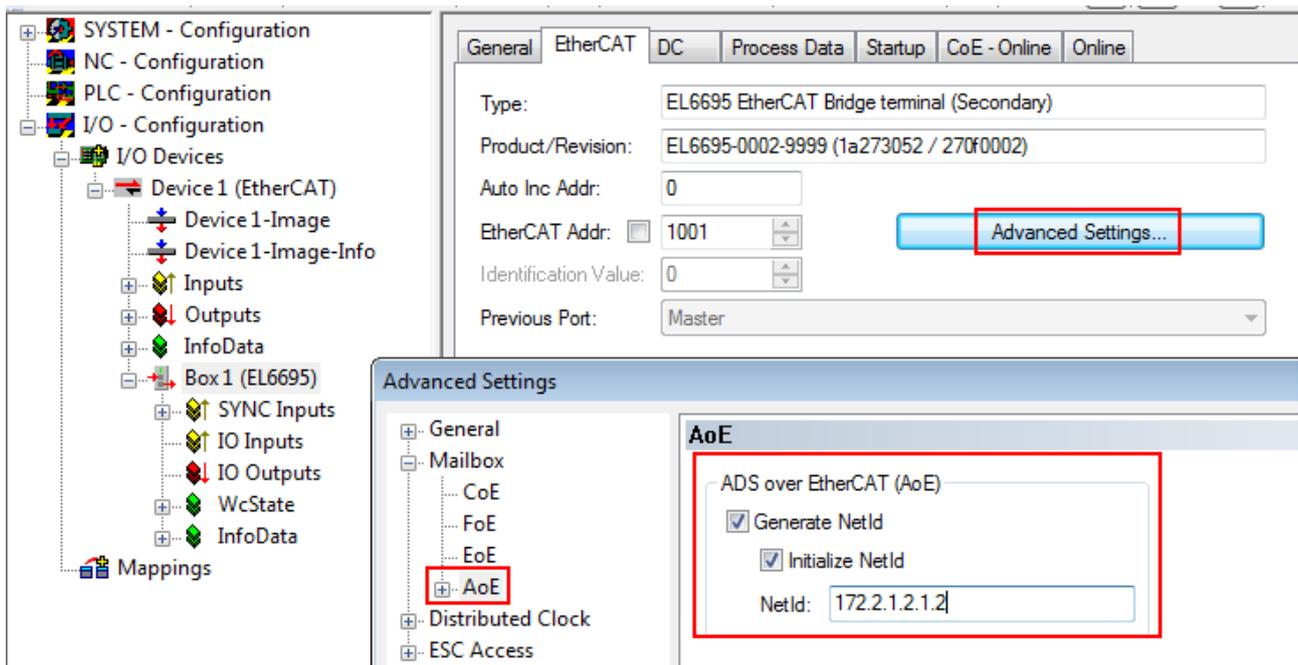


図 153: NetIDの採用

i ● 必要なAMSアドレスの格納

他のデバイスによってオブジェクトディクショナリ (OD)の一部を交換しODに追加される部分的なODを使用する場合、対応するAMSアドレスが各(追加の)オブジェクトに対して格納されている必要があります(オブジェクト0xF820 [▶ 104]を使用)。

- ・モード「部分的なオブジェクトディレクトリ」：
 - ODの一部がダウンロードされ、ODに挿入されます。
- ・モード「ミラーリングされるオブジェクトがある部分的なオブジェクトディレクトリ」：
 - ODの一部がEL6695にロードされ、ODに挿入されます。加えて、ミラーリングされたオブジェクトがリモートOD内に作成されます。このモードにより、前述AoEメカニズムを使用せずに、CoE AとCoE B間の直接データリンクが可能になります。ただし、このモードはパラメータ範囲0x8nnnしかカバーしません。
 - CoE A 0x8000 - 0x87FFをCoE B 0x8800 - 0x8FFFにミラーリング。
 - CoE B 0x8000 - 0x87FFをCoE A 0x8800 - 0x8FFFにミラーリング。
 - このため、0x88xyオブジェクトは「読み取り専用」(RO)となります。

i ● 変更したCoEオブジェクト

- 元の名前は、CoEオブジェクト0x10E1「OEM Device Name」に文字列「Beckhoff EL6695」として格納されています。これは変更できません。
- ODは、0x1C13以降を使用してプロセスデータも定義します。ソース側とエミュレートする側のプロセスデータ長が同一である必要があります。
- ODはEL6695内に電源故障時でも安全な状態で格納されます。最大サイズは片側128kbytesです。
- マスタBのエミュレートする側にアクセスするには、ソース側がオンラインであるか、存在しており、EL6695の動作を定義している必要があります。Bが先に開始すると、EL6695はPreOp_ERRで応答し、プロセスデータが未定義であることを示します。マスタAが開始され、EL6695を定義すると、マスタBは「ERRのクリア」を実行し、OP時にB側を再起動する必要があります。動作はCoEオブジェクト0xF800:02のビット1 (MASK_RUN_PRI_UNC)およびビット2 (MASK_RUN_SCND_UNC)を使用して設定できます(付録を参照)。

- ・モード「完全」、「部分的」、「ミラーリング」は、EL6695内の*.coeファイルの内容で設定できません。

6.6.2 EoE – Ethernet over EtherCAT

処理：転送

プライマリ/セカンダリで異なるメールボックスサイズのサポート

TwinCAT、および接続されているEtherCAT環境は仮想ネットワークカードとして動作するため、WindowsはIPフレームのルーティング(192.168.2.1など)を処理して、他方のEtherCATシステムにフレームを送信します。基本原則：EL6601/EL6614の詳細マニュアルを参照してください。

EL6695内でEoEを有効にするには、EoEの[Advanced Settings]にあるターミナルの[EtherCAT]タブ内で[Virtual Ethernet Port]オプションにチェックマークを入れ、[Switch Port]を選択します。

The screenshot displays the configuration interface for an EL6695 EtherCAT Bridge terminal. The left sidebar shows a project tree where 'Term 10 (EL6695)' is selected. The main window shows the 'EtherCAT' configuration tab with the 'Advanced Settings...' button highlighted. The 'Advanced Settings' dialog is open, showing the 'EoE' section where the 'Virtual Ethernet Port' checkbox is checked and the 'Switch Port' radio button is selected. The 'Virtual MAC Id' is set to '02 01 05 10 03 eb'. Other options like 'IP Port', 'DHCP', 'IP Address', 'Subnet Mask', 'Default Gateway', 'DNS Server', and 'DNS Name' are also visible but not selected.

図 154: EoEコンフィグレーション

● i オブジェクト0xF800 (Device Config)を使用したEoEの有効化

必要に応じて、ターミナル0xF800のCoEオブジェクト(Device Config)を使用して、EoEが有効であるかどうかをチェックします。エン트리0xF800:01 (Config 1)内でビット14が設定されている(値0x4000などが入力されている)場合、EoEメールボックスプロトコルはブロックされます(付録を参照)。有効にするには、このビットを0に設定するか、16ビット値を完全に0x0000に設定します。または、スタートアップエントリからこれを行うことも可能です(図 [▶ 129]を参照)。

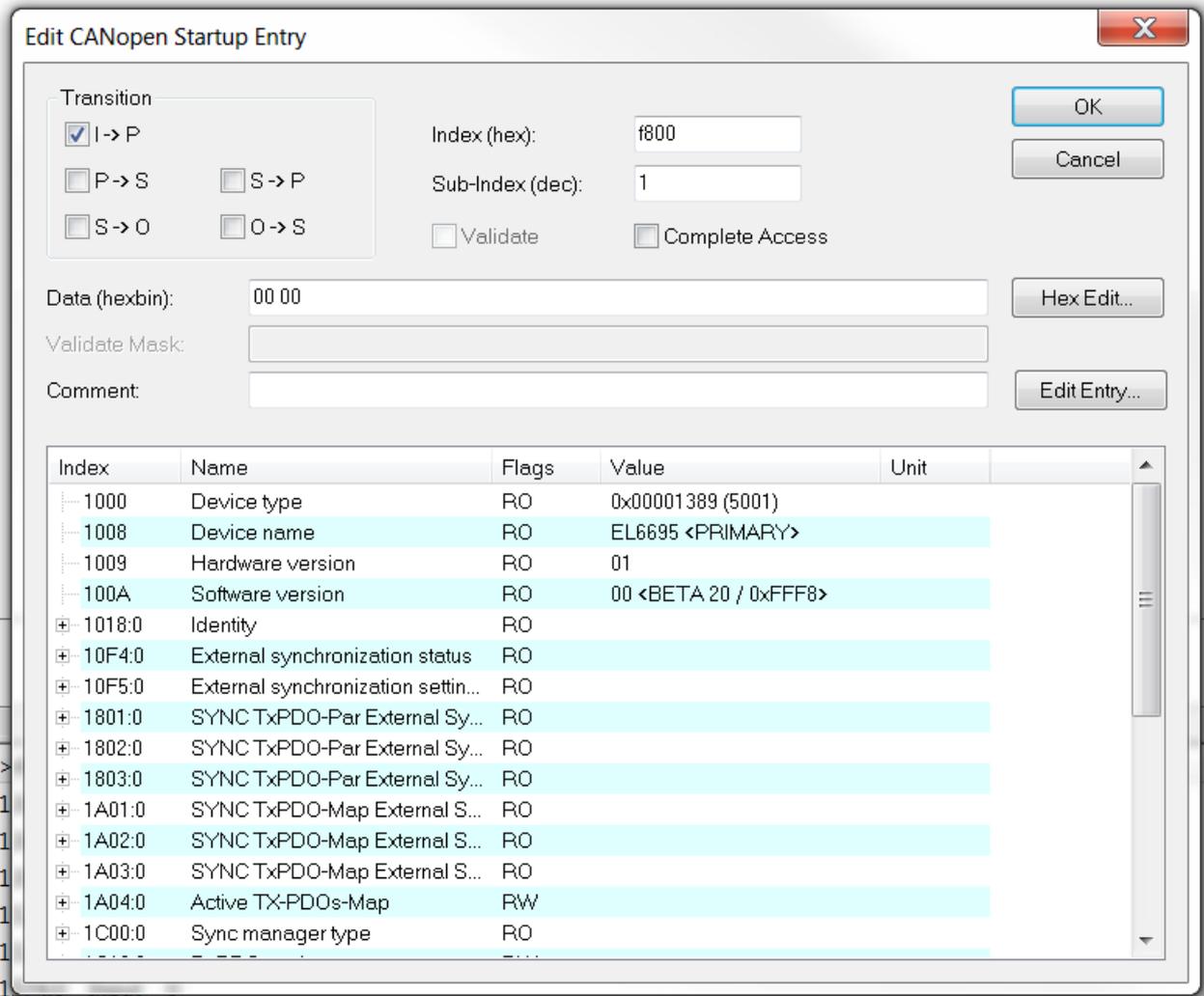


図 155: オブジェクト0xF800の正しい値 (0x0000) の検証: EoE有効

EoE通信を動作させるには、両方のEtherCATマスタのIPアドレスが同一サブネット内に存在している必要があります。

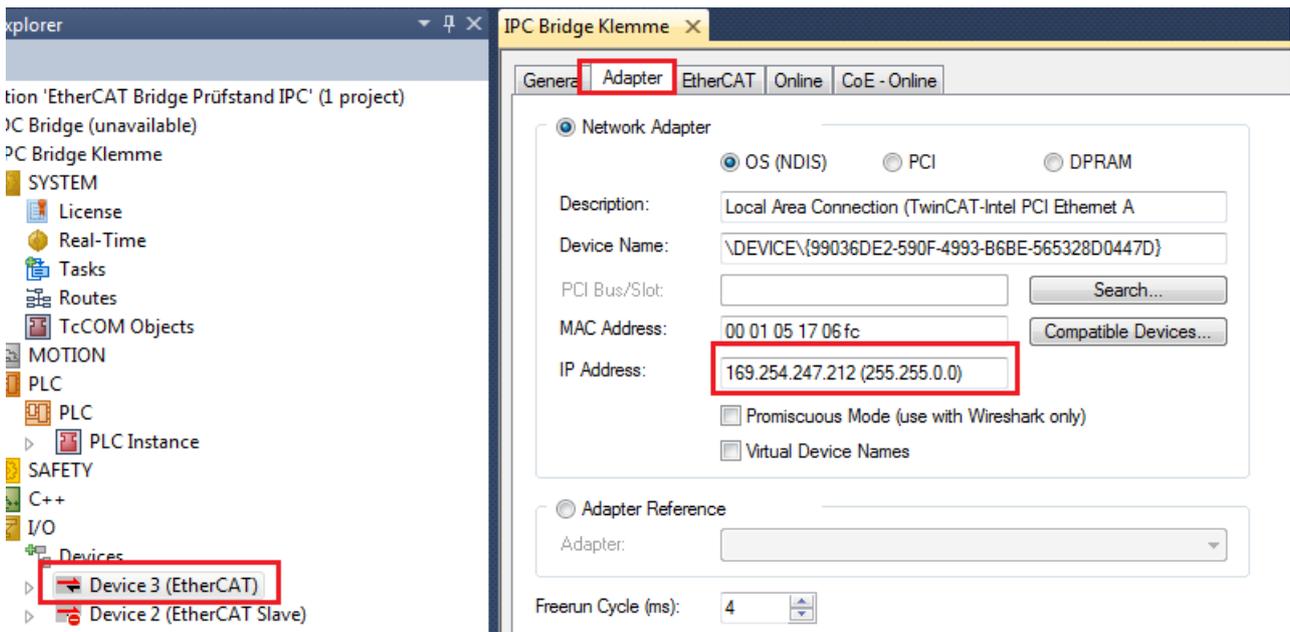


図 156: EoE IPアドレスのチェック

この例では、マスタ1のIPアドレスは169.254.247.212、サブネットは255.255.0.0です。

マスタ2のIPアドレスは例えば169.254.249.204とすることができますが、サブネット255.255.0.0内に存在する必要があります。

この場合、マスタ1のコマンドラインで「ping」コマンドを発行すると、マスタ2にコンタクトできます。

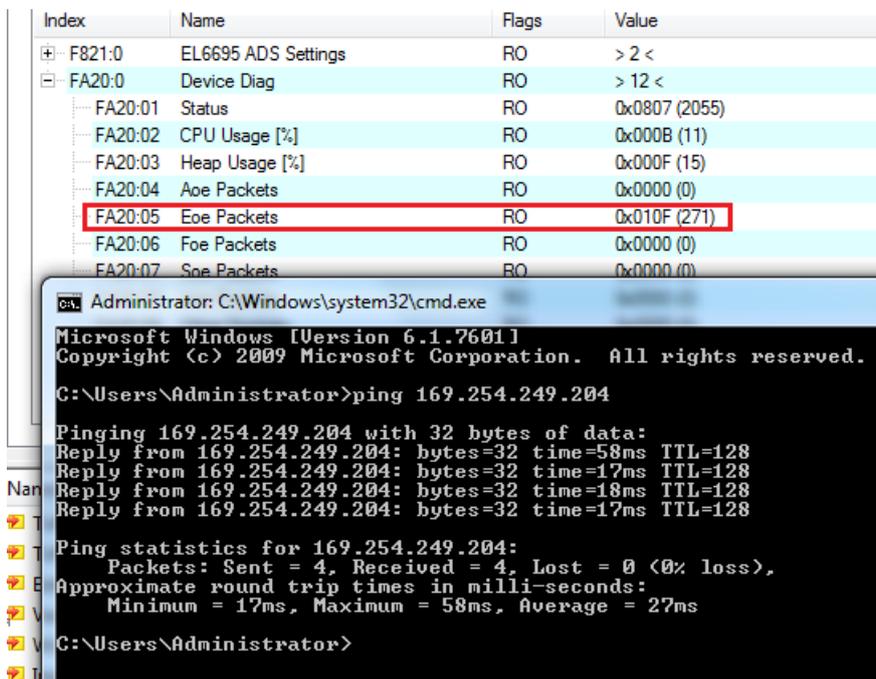


図 157: コマンドラインを使用したマスタ1からマスタ2への「ping」の実行

EL6695のCoEオブジェクトでは、オブジェクト0xFA20:05内でEoEパケットがインクリメントされます。この例では、pingコマンドを使用して4つのパケットを送信します。

6.6.3 AoE – ADS over EtherCAT

AoE – ADS over EtherCAT

処理: 転送

プライマリ/セカンダリで異なるメールボックスサイズのサポート

ADS (Automation Device Specification)は、ハードウェアまたはソフトウェアベースのデバイス間のデータ交換用にベッコフが開発し、公開しているプロトコルです。ADSテレグラムの構造は、Beckhoff Information SystemまたはETG規格で確認できます。EL6695は下層のIPチャンネルを使用せず、AoE (ADS over EtherCAT)によってADSテレグラムを他方に直接送信できます。これを行うには、AoEによる通信が両側で有効になっている必要があります。

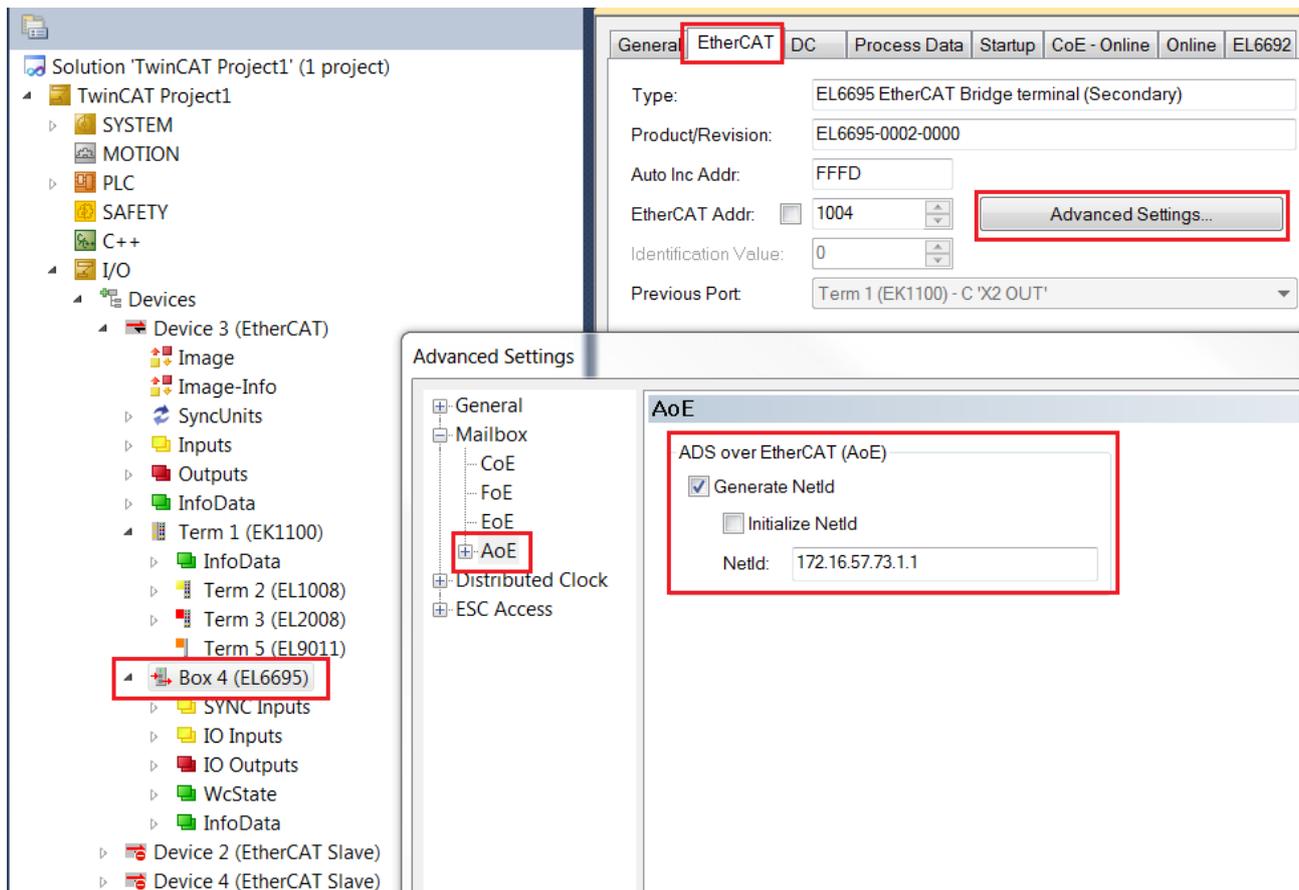


図 158: ADSフレームルーティングの設定

ADSフレームをどのように、またはどのチャンネルを使用してルーティングするかをTwinCATに通知するために、他のシステムのアドレスをNetIDとして設定する必要があります。上図はPC2の場合の例です。

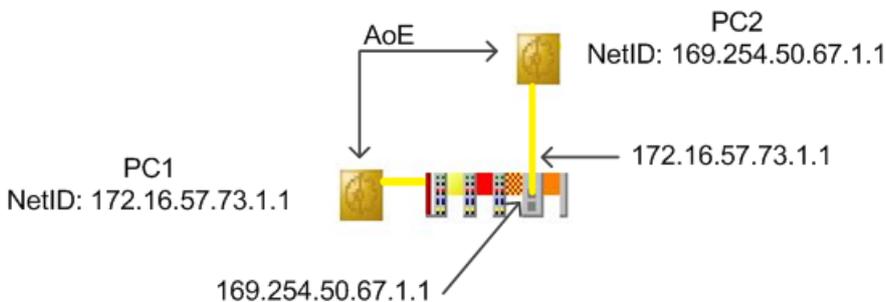


図 159: AoE (ADS over EtherCAT)用のNetIDの設定

● サンプルプログラムの使用

i 本取扱説明書には、当社製品の特定の用途領域向けのサンプルアプリケーションが含まれていません。ここで提供されているアプリケーションノートは、当社製品の一般的な特徴に基づいたものであり、サンプルとしてのみお使いいただけます。本取扱説明書に含まれているアプリケーションノートは、決して特定のアプリケーションを対象としたものではありません。このため、お客様の責任の下、特定のアプリケーションに対して製品が適しているかどうかを評価および判断してください。当社は、本取扱説明書に含まれているソースコードの完全性および正確性に対して、いかなる責任も負いません。当社は、本取扱説明書の内容を随時変更する権利を有する一方で、誤りおよび情報の不足については一切の責任を負いません。

→ ご要望により、サンプルプログラムをご提供します。

<https://infosys.beckhoff.com/content/1033/el6695/Resources/zip/1421139595.zip>

6.6.4 FoE - Filetransfer over EtherCAT

FoE - Filetransfer over EtherCAT

本来FoEは、関与している各マスタ (PLC) の2つの参照されるメモリ領域を使用したデータアクセスを可能にするためのものです。データはプライマリ (Eバス側) からセカンダリ (RJ45接続) へ、またはセカンダリからプライマリへストリーム形式で送信できます。

各メモリの最大サイズは32768バイトです。

EtherCAT状態ごとのEL6695内での処理:

- ・ BOOT: FoEによりローカルのFW更新を行います。
- ・ その他すべての状態 (INIT、PreOP、SafeOP、OP): CoEパスワードの有無に応じて異なります。
 - パスワードなし: FoEがリモートサイトに転送されます。
 - パスワードあり: パスワードに応じて、ローカルで保存されます。これにより、特殊機能が提供されます。

● FoEリクエスト/パスワードなし

I FoEリクエストは、反対側によって承認され、処理する必要があります。TwinCATバージョン3.10 b4014および2.11 b2245は、これをサポートしていません。メールボックスサイズは、両側で同一である必要があります。

バッファリングFoE動作モード

オブジェクト0xF800 (Device Config)、サブインデックス02で、[MASK_BUFFER_FOE] = ビット8が設定されている場合は、異なるサイズのメールボックスを使用できます。

6.6.4.1 TwinCATでのFoEの要件

TwinCATでのプログラミング

プログラミングではライブラリ「Tc_EtherCAT」とEtherCATマスタおよびスレーブデバイスへのアクセス用のファンクションブロックを使用する必要があります (TwinCAT 3では「Tc2_...」が利用可能)。TwinCATに含まれている以下のファンクションブロックにより、データやファイル通信機能にアクセスが可能になります。

- ・ FUNCTION_BLOCK FB_EcFoeOpen (通信ポートを開く)
- ・ FUNCTION_BLOCK FB_EcFoeAccess (データの書き込み/読み取り)
- ・ FUNCTION_BLOCK FB_EcFoeClose (通信ポートを閉じる)

詳細情報は、ベッコフの情報ページ (infosys.beckhoff.com) を参照してください。

注意: FoE転送は、ファイルが正常に終了し、データ送信が完了したかどうかを自身ではチェックできません。ファイル自体に検証可能なメタ情報などを挿入するといった他の方法でファイルサイズまたはファイルの完全性を送信することを推奨します。

EL6695によるFoEデータ交換には、以下のパターンがあります。

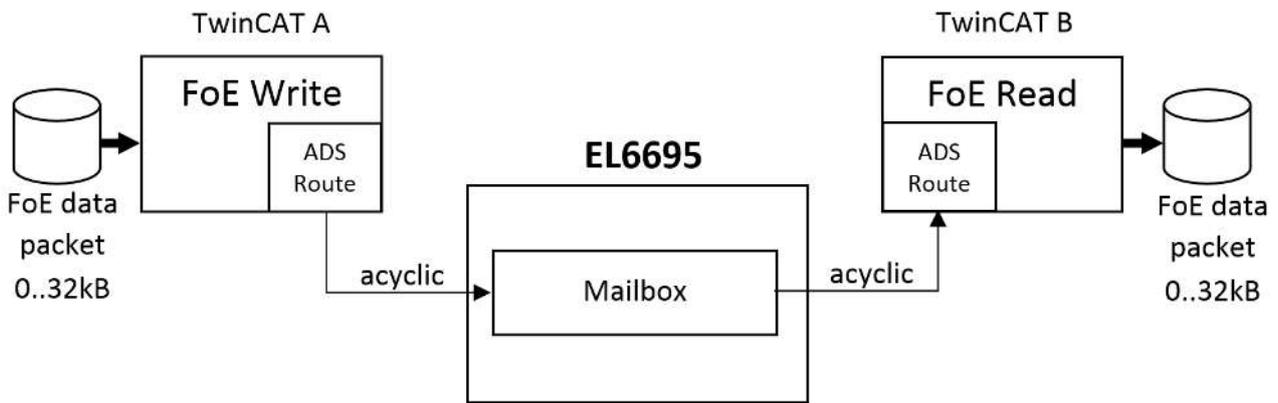


図 160: EL6695によるFoEデータ交換

- ・ FoEデータセット、1セットの最大サイズ32 kB、EL6695によって送信可能
- ・ EL6695メールボックス(デフォルト1024バイト)による非周期的EtherCAT通信を使用
- ・ 自身のFoE処理がない場合、EL6695は単なるデータバッファとして機能

6.6.4.2 コンフィグレーション例

コンフィグレーション例

ここでは、プログラミング例を使用してFoEの機能を解説します。下図 [▶ 134]は、CX2100-0014電源ユニットおよび下流のEL2809、EL1004、およびEL6695を含む、CX2040組込み型PCの物理構造を示しています。ここでは、CX PCのイーサネットコネクタX001が、EL6695ブリッジターミナルの「上側」のRJ45コネクタX1に接続されています。

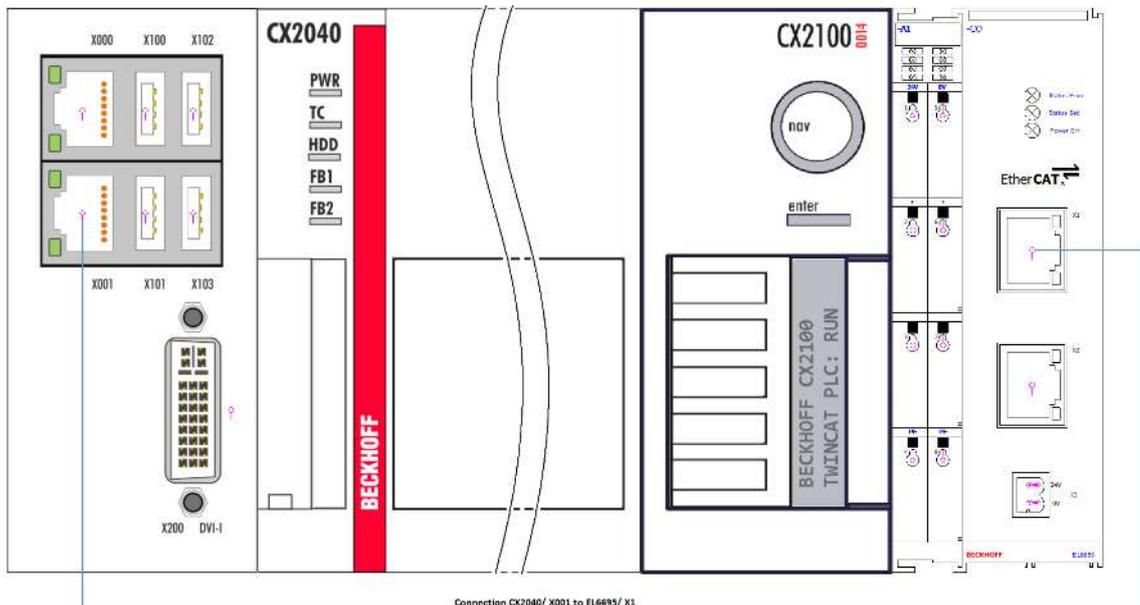


図 161: CX2100-0014電源ユニットおよびEL6695を含むCX2040での構成

さらに、情報オブジェクト「Data Bytes Pending」に基づいて、逆側でデータまたはファイルが受信されたかどうかを検出できるように、CoEオブジェクト0x1A05をSystem Managerに追加されます。これを行うには、対応するチェックボックスがプロセスデータ内で有効になっている必要があります。この例では、逆側として示されているセカンダリでのみ有効になっている状態が表示されています。さらに、TwinCATサンプルプログラムに含まれているオープンにアクセス可能な変数「ScndFoeBytesToRead」は、「Data Bytes Pending」にリンクします。

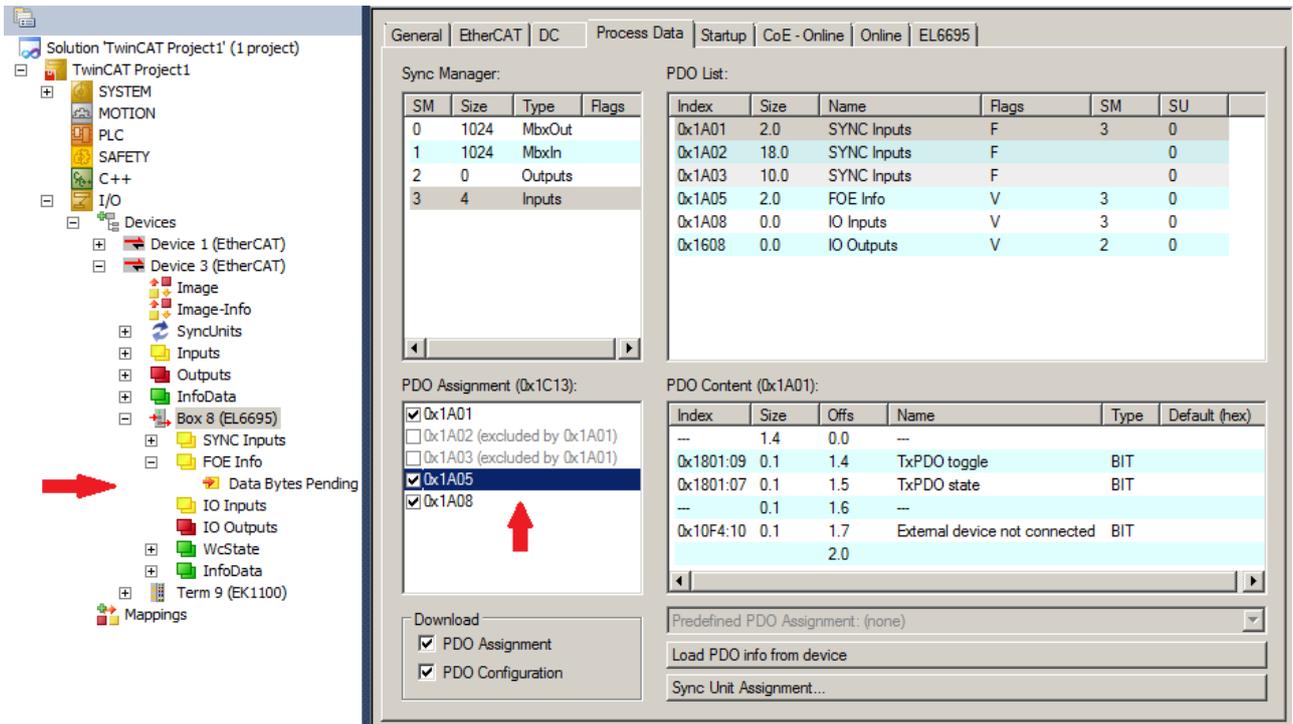


図 162: セカンダリへの0x1A05 (Data Bytes Pending)の追加

Init - スタートアップコンフィグレーション

オブジェクト0xF800のコンフィグレーションに基づき (CoE/パラメータディレクトリ - プロファイル固有のオブジェクトを参照)、Init状態からPreOp状態 (I → P) に遷移させるためには0xF800に2つの値を入力する必要があります。

- ・ 0xF800:01の値0x4000 (EoEを無効化)
- ・ 0xF800:02の値0x0100 (EL6695上にFoEバッファメモリを確保するための特殊モード)

手順は下図の、System Manager (TC3.1)のユーザインターフェイス上にある[Startup]タブのプライマリまたはセカンダリのマークが付けられたターミナルまたはボックスで示しています。

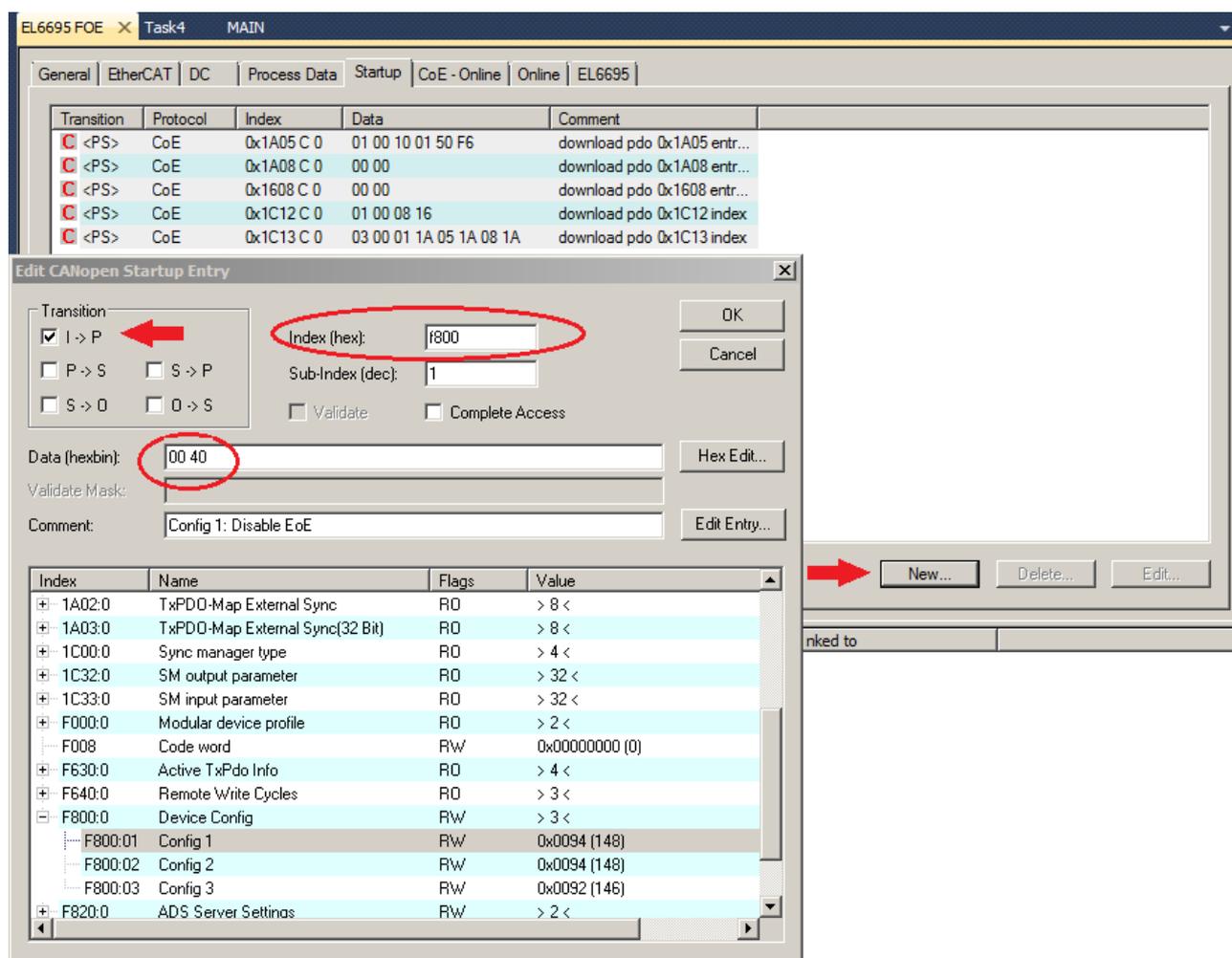


図 163: デバイスコンフィグレーション0xF800:01内に値0x4000を入力(コメント = Config 1: Disable EoE)

ここではEoEを無効にするために、オブジェクト0xF800、サブインデックス01に値0x4000 (タイプ00 40)を入力します。この例では両側が1つのEtherCATマスタに接続されているため、この値の入力はターミナルのプライマリとセカンダリを接続するためにだけ必要です。

● FoEでのEoEプロトコルのブロック

I プライマリおよびセカンダリが1つのEtherCATラインに接続されている場合、ターミナルでオブジェクト0xF800:01にビット14 (0x4000)を設定してEoEプロトコルをブロックする必要があります。これを行わないとARPイーサネット要求が繰り返され、ターミナルがブロックされます。

さらに、0xF800:02には値0x0100 (タイプ00 01)を入力する必要があります(コメント = Config 2: Enable FoE Buffer)。これらの設定は必ず逆側にも適用されるため、EL6695ブリッジターミナルのプライマリとセカンダリのどちらで行っても関係ありません。

● Device Config 0xF800の変更

I オブジェクト0xF800は、プライマリとセカンダリの両方で設定変更が可能であり、必ず逆側にも適用されます。

正しい遷移が選択されているかどうかには注意が必要です。P → Sはオフになっている、I → Pは有効になっている必要があります。この設定が完了すると、下図のようになります。

General EtherCAT DC Process Data Startup CoE - Online Online EL6695					
Transition	Protocol	Index	Data	Comment	
C <PS>	CoE	0x1A05 C 0	01 00 10 01 50 F6	download pdo 0x1A05 entries	
C <PS>	CoE	0x1A08 C 0	00 00	download pdo 0x1A08 entries	
C <PS>	CoE	0x1608 C 0	00 00	download pdo 0x1608 entries	
C <PS>	CoE	0x1C12 C 0	01 00 08 16	download pdo 0x1C12 index	
C <PS>	CoE	0x1C13 C 0	03 00 01 1A 05 1A 08 1A	download pdo 0x1C13 index	
C IP	CoE	0xF800:01	0x4000 (16384)	Config 1: Disable EoE	
C IP	CoE	0xF800:02	0x0100 (256)	Config 2: Enable FoE Buffer	

図 164: デバイスコンフィグレーション0xF800に両方の値を入力

プログラム例に関する説明

FoEデータ送信を説明するための付属のプログラム例は、以下に示す書き込みおよび読み取りアクセスの状態図に基づいています。

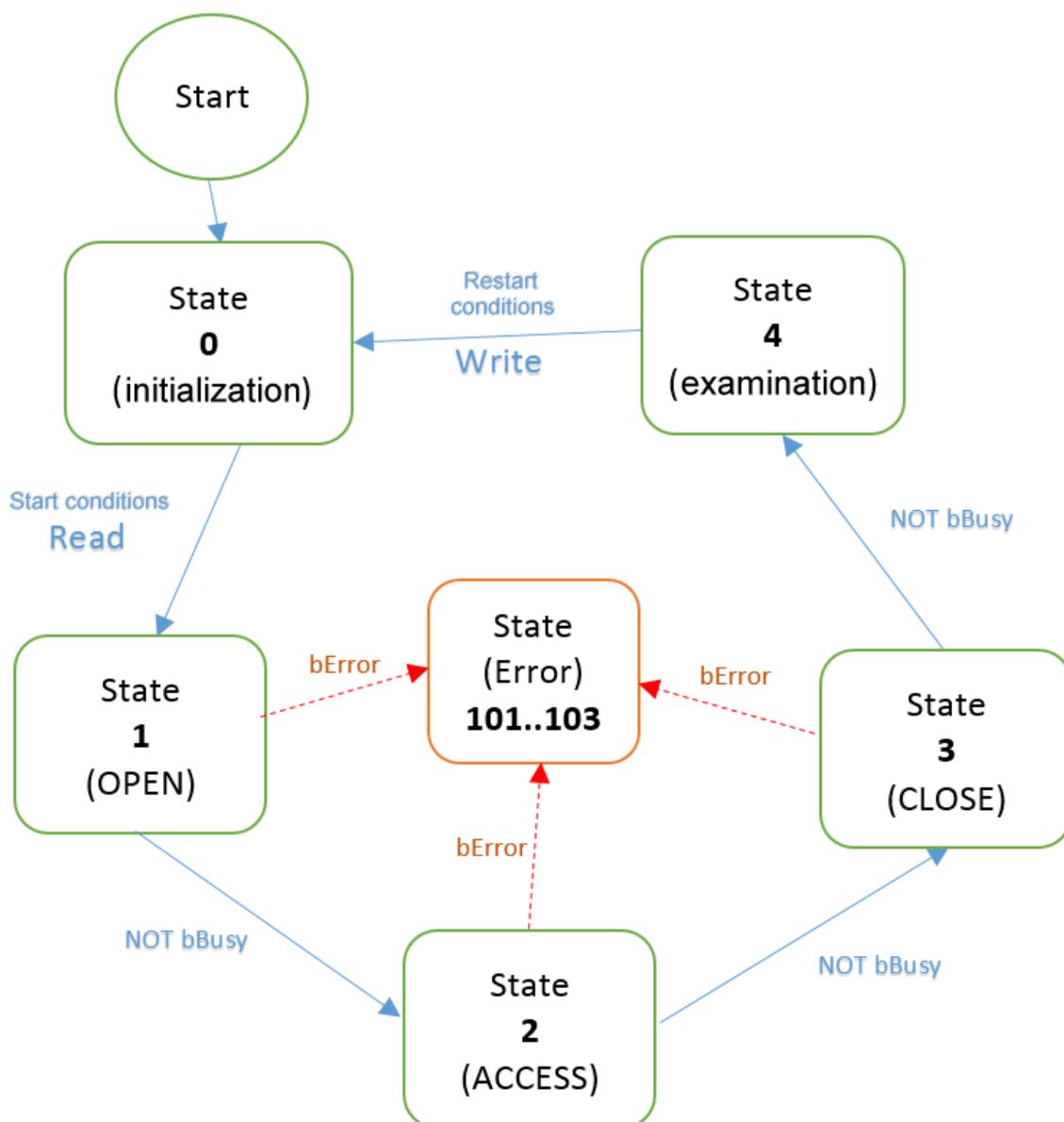


図 165: プログラム例の状態図: 「OPEN」、「ACCESS」、および「CLOSE」によるランダムデータの書き込みおよび読み取り

プログラム内で定義されている**bEnabled**フラグは、プログラムを制御し、開始条件や再開条件として使用するためのものです。追加宣言AT%I*により、このフラグを入力ターミナルの「実際の」入力とリンクし、接続されたボタン(上+)などで「外部から」プログラムを制御することが可能です。

- ・ 開始条件Read: **bEnabled**がTRUEである必要があります(外部接続されたボタンの起動など)。さらに、受信データに対して「ScndFoeBytesToRead」が0よりも大きな値であるかどうかをチェックされます。
- ・ 再開条件Write: **bEnabled**が一度FALSEとなっている必要があります(外部接続されたボタンの解放など)。

さらに、データ書き込みに対してステートマシンの状態1内で**bEnabled**フラグがTRUEであるかどうかをテストされ、TRUEであった場合のみ、OPENによって書き込み処理を開始します。

追加宣言AT%Q*により、デジタル出力を提供するターミナルの出力変数と**bDataEqual**をリンクできます。これにより、正常なデータ送信が行われたかどうかを判別できます。

個々の状態0~4は、**iWrState**と同様にプログラムされます。

- ・ [**iWrState =0**]: 初期化の実行:
 - FALSEをセット → **bDataEqual**
 - **aWrBuffer**でのランダムデータ生成

- OPENのファンクションブロックコールの準備: FALSE → **bExecute**をセット
- 次の状態をセット: 1 → **iWrState**
- **[iWrState =1]:** OPENのファンクションブロックのコールを開始するかどうかを入力変数**bEnabled**に照会します。**bEnabled** = TRUEの場合、OPENファンクションブロックを以下のパラメータでコールします。
 - EC_MasterNetId_Wr → **sNetId**
 - EL6695_WR_EcAddr → **nPort**
 - 16#12345678 → **dwPass**
 - 1 → **eMode** (書き込みの識別子)
 - TRUE → **bExecute**
 - T#10S → **tTimeout**
 - このコールは、**bBusy**フラグがFALSEを返すまで繰り返されます。その後、以下が行われます。
 - 次の状態をセット: 2 → **iWrState**。 **bBusy** = TRUEの場合、状態は変化しません。OPENファンクションブロック (読み取り/書き込み用) の**bError**フラグによるエラー時の処理は、この例、およびその他のファンクションブロック (読み取り/書き込み用のACCESS、CLOSE) では考慮されていません。
 - ACCESSのファンクションブロックコールの準備: FALSEをセット → **bExecute**
- **[iWrState =2]:** ACCESSファンクションブロックを以下のパラメータでコールします。
 - fbWrFoeOpen. hFoe → **hFoe**
 - ADR(aWrBuffer) → **pBuffer**
 - TRUE → **bExecute**
 - T#14S → **tTimeout**
 - このコールは、**bBusy**フラグがFALSEを返すまで繰り返されます。その後、以下が行われます。
 - 次の状態をセット: 3 → **iWrState**
 - CLOSEのファンクションブロックコールの準備: FALSE → **bExecute**をセット
- **[iWrState =3]:** CLOSEファンクションブロックを以下のパラメータでコールします。
 - fbWrFoeOpen. hFoe → **hFoe**
 - TRUE → **bExecute**
 - T#14S → **tTimeout**
 - このコールは、**bBusy**フラグがFALSEを返すまで繰り返されます。その後、以下が行われます。
 - 次の状態をセット: 4 → **iWrState**
- **[iWrState =4]:** この状態では、入力変数 **bEnabled**が再度FALSEとなっているかどうかだけを照会します。FALSEの場合、プログラムは開始状態から再度実行します。
 - **bEnabled** = FALSEの場合、次の状態をセット: 0 → **iWrState**

次の開始では、新しいランダムの数値が新たに生成されます。これらの数値は読み取り値と一致する必要があります。後述するように、これらの数値が一致しているかどうかを示すために**bDataEqual**フラグが用意されています。

読み取りアクセスの状態図は原則どおりとなり、状態4には書き込み値と読み取り値のデータ比較用のプログラムセクションが設けられます。

個々の状態0~4は、**iRdState**と同様にプログラムします。

- **[iRdState =0]:** この状態では、入力変数**bEnabled**がTRUEとなっているかどうかを照会します。これにより、プログラムは[FoE Info]. Data Bytes Pending (オブジェクト0x1A05)とリンクされた入力変数「ScndFoeBytesToRead」の照会を開始し、データが受信側に「存在する」かどうかを判定します。
 - **bDataEqual** = TRUEの場合、**ScndFoeBytesToRead**の照会
 - **ScndFoeBytesToRead** = TRUEの場合、次の状態をセット: 1 → **iRdState**
 - OPENのファンクションブロックコールの準備をセット: **ScndFoeBytesToRead** = TRUEの場合、FALSE → **bExecute**をセット
- **[iRdState =1]:** OPENのファンクションブロックのコールを開始するかどうかを入力変数**bEnabled**に照会します。**bEnabled** = TRUEの場合、OPENファンクションブロックを以下のパラメータでコールします。
 - EC_MasterNetId_Rd → **sNetId**
 - EL6695_RD_EcAddr → **nPort**
 - 16#12345678 → **dwPass**
 - 0 → **eMode** (読み取りの識別子)
 - TRUE → **bExecute**

- T#10S → tTimeout
- このコールは、bBusyフラグがFALSEを返すまで繰り返されます。その後、以下が行われます。
- 次の状態をセット: 2 → iRdState
- ACCESSのファンクションブロックコールの準備: FALSEをセット → bExecute
- ・ [iRdState =2]: ACCESSファンクションブロックを以下のパラメータでコールします。
 - fbRdFoeOpen. hFoe → hFoe
 - ADR(aRdBuffer) → pBuffer
 - TRUE → bExecute
 - T#14S → tTimeout
 - このコールは、bBusyフラグがFALSEを返すまで繰り返されます。その後、以下が行われます。
 - 次の状態をセット: 3 → iRdState
 - CLOSEのファンクションブロックコールの準備: FALSE → bExecuteをセット
- ・ [iRdState =3]: CLOSEファンクションブロックを以下のパラメータでコールします。
 - fbRdFoeOpen. hFoe → hFoe
 - TRUE → bExecute
 - T#14S → tTimeout
 - このコールは、bBusyフラグがFALSEを返すまで繰り返されます。その後、以下が行われます。
 - 次の状態をセット: 4 → iRdState
- ・ [iRdState =4]: この状態では、終端で両方のメモリ領域のチェックを行い、両方のメモリ領域が同一の場合はbDataEqualフラグをTRUEにセットします。このフラグは、出力変数として「可視化」しておくことを推奨します。これにより、出力ターミナルによる出力への切り替えなどが可能になります。
 - 次の状態をセット: 0 → iRdState

● サンプルプログラムの使用

i 本取扱説明書には、当社製品の特定の用途領域向けのサンプルアプリケーションが含まれています。ここで提供されているアプリケーションノートは、当社製品の一般的な特徴に基づいたものであり、サンプルとしてのみお使いいただけます。本取扱説明書に含まれているアプリケーションノートは、決して特定のアプリケーションを対象としたものではありません。このため、お客様の責任の下、特定のアプリケーションに対して製品が適しているかどうかを評価および判断してください。当社は、本取扱説明書に含まれているソースコードの完全性および正確性に対して、いかなる責任も負いません。当社は、本取扱説明書の内容を随時変更する権利を有する一方で、誤りおよび情報の不足については一切の責任を負いません。

→ サンプルプログラムFoEのダウンロード:

<https://infosys.beckhoff.com/content/1033/el6695/Resources/zip/1421822987.zip>

サンプルプログラム (tnzipファイル/TwinCAT 3) 開始の準備

- ・ [Download] ボタンをクリックしてローカルのハードディスクにZipアーカイブを保存してから、一時フォルダ内で*.tnzipアーカイブを解凍します。

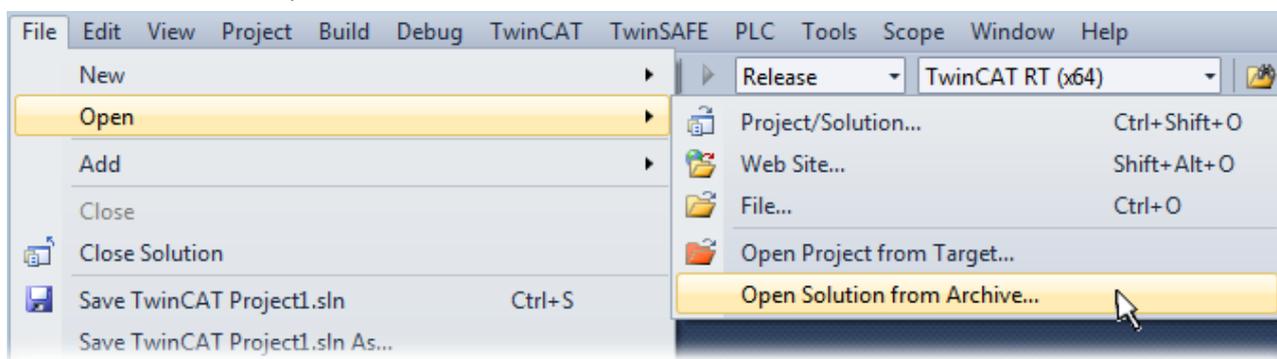


図 166: *.tnzipアーカイブを開く

- ・ .tnzipファイル(サンプルプログラム)を選択します。
- ・ 次の選択ウィンドウが開きます。プロジェクトを保存する宛先ディレクトリを選択します。
- ・ 一般的なPLCのコミッショニング手順およびプログラムの開始に関する説明は、ターミナルの取扱説明書またはEtherCAT System Documentationを参照してください。

- ・ サンプルのEtherCATデバイスは、通常、現在のシステム上で宣言する必要があります。Solution ExplorerでのEtherCATデバイスの選択後、[Adapter]タブを選択して[Search...]をクリックします。

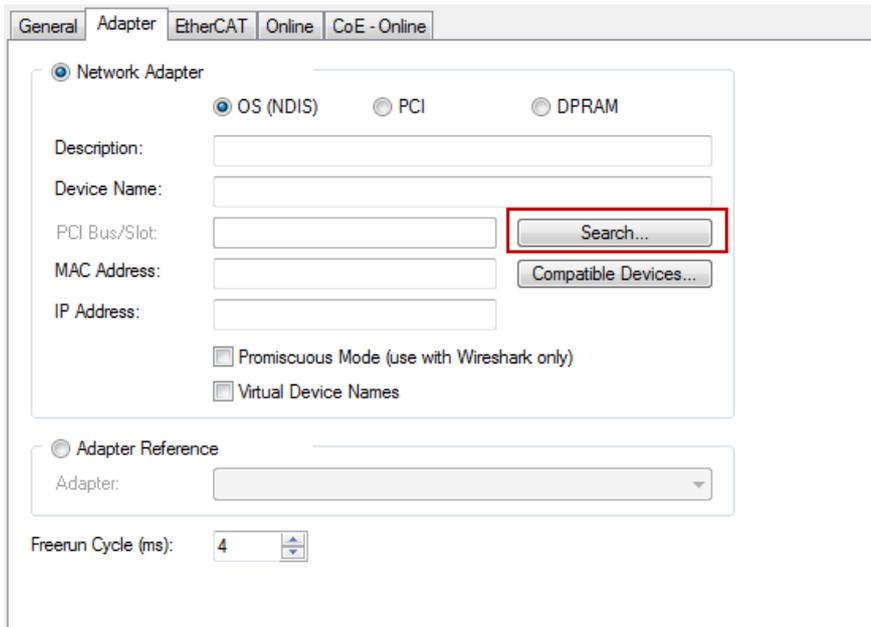
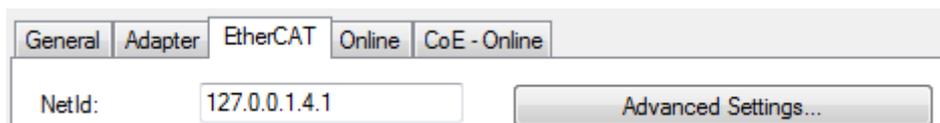


図 167: サンプルのEtherCATコンフィグレーションで既存のHWコンフィグレーションを検索

- ・ NetIdのチェック: EtherCATデバイスの[EtherCAT]タブに、設定済みのNetIdが表示されます。



先頭の4つの数値が、ターゲットシステムのプロジェクトNetIdと同一である必要があります。プロジェクトNetIdは、上部のTwinCAT環境内でプルダウンメニューを開き(テキストフィールド内を右クリック)、ターゲットシステムを選択することで表示できます。ここでは、数値ブロックがターゲットシステムの各コンピュータ名の隣にブラケットで囲まれて表示されます。

- ・ NetIdの変更: Solution Explorer内の[EtherCAT device]を右クリックすると、コンテキストメニューが開きます。ここで[Change NetId...]を選択します。ターゲットコンピュータのNetIdの先頭の4つの数値を入力する必要があります。通常、最後の値は両方とも4.1となります。

例:

- プロジェクトのNetId: myComputer (123.45.67.89.1.1)
- [Change NetId...]で入力: 123.45.67.89.4.1

6.6.4.3 サンプル: FoEデータスループット

以下は、EL6695を使用したFoEスループット判定のサンプルです。値はこのレイアウトの例であり、大体の目安です。各アプリケーションで実現可能な実際のスループットは、必要に応じて別途判定する必要があります。

これに使用するTwinCATプログラムは、本取扱説明書の範囲外です。これは、指定したFoEサンプルプログラム [\[▶ 140\]](#)に基づきます。

データスループットは、データ量と必要な時間との比率で定義されます(時間単位当たりのデータ量)。前述のFoEアクセスブロックを使用すると、一定のデータ量がEL6695へ書き込まれる、またはEL6695から読み込まれます。EL6695は、2つの異なるTwinCATシステム間の通信を確立するように設計されています。ただし、TwinCATシステムのX1を経由するセカンダリがEK100カブラの出力X2から接続されている場合など、ターミナルでTwinCATシステム内のプライマリとセカンダリの接続を確立することも可能です。これにより、TwinCATシステム内に2つの独立したEtherCATセグメントができます(参考: [「コンフィグレーション例 \[▶ 134\]」](#))。

時間計測は、FB_EcFoeAccessファンクションブロックを使用して読み取りアクセスに要したタスクサイクル数(書き込まれたバイト数は読み取られたバイト数と同一)によって間接的にを行います(「[プログラム例に関する注記 \[▶ 137\]](#)」を参照)。データが保留中の場合、読み取られるターミナル側の「DataBytesPending」PDOは0よりも大きな値となり、読み取りアクセスの開始を示します。読み取りアクセスの終了は、既知の書き込みデータ量から判定されます(2つの独立したシステムではファイルサイズを認識する必要があります)。

タスクサイクルタイムが1 msの場合、データスループットは単純に以下のように計算されます。
[データ量[バイト]] / [タスクサイクル数] = スループット[kB/s]

計測手順

- ・ テスト環境では、ターミナルが読み取りアクセスに必要なタスクサイクル数を判定しました。ターミナルに書き込まれたデータパケット、およびデータ自体のサイズがさまざまなデータセットを、繰り返しかつ完全に再度読み出されました。これによって、読み取りアクセスが発生するたびにタスクサイクルをカウントします。
- ・ EL6695を間に接続しない場合のテストプログラムのFoEスループットも測定します。これが、テストプログラムの「実質的な」時間要件となります。
- ・ ターミナルが必要とする実際のタスクサイクル数(およびデータスループット)は、EL6695に必要な「実際の」タスクサイクルと実質上のタスクサイクルの中間の数となります。
- ・ スループットテスト全体では、さまざまなデータ量およびデータパケットサイズで数千回のテストを行います。

サンプル内のコンフィグレーション:

- ・ C6015 + EK1100 + EL6695 (プライマリ) + EL9011
- ・ EK1100 + EL6695-0002 (セカンダリ)

サンプルの主要データ:

- ・ サイクルタイム = 1 ms
- ・ テスト実行時にFoEのさまざまなパケットサイズをテストした結果、スループットは定義された主要範囲を大きく超えて変化することはありません。したがって、以下の圧縮された画像に対して次の制限値を選択します。
- ・ FoEが送信するデータ量 (FoEファイルサイズ): 50ステップで1167~32767バイト、各ステップで632バイト
- ・ データパケット (FoEファイル分割後のパケット): 40ステップで30~470バイト、各ステップで11バイト

結果:

データパケットサイズ(ステップ内でインクリメント) = X軸

データ量(ステップ内でインクリメント) = Y軸

必要なタスクサイクル数 = Z軸、Z(X, Y)と等しい

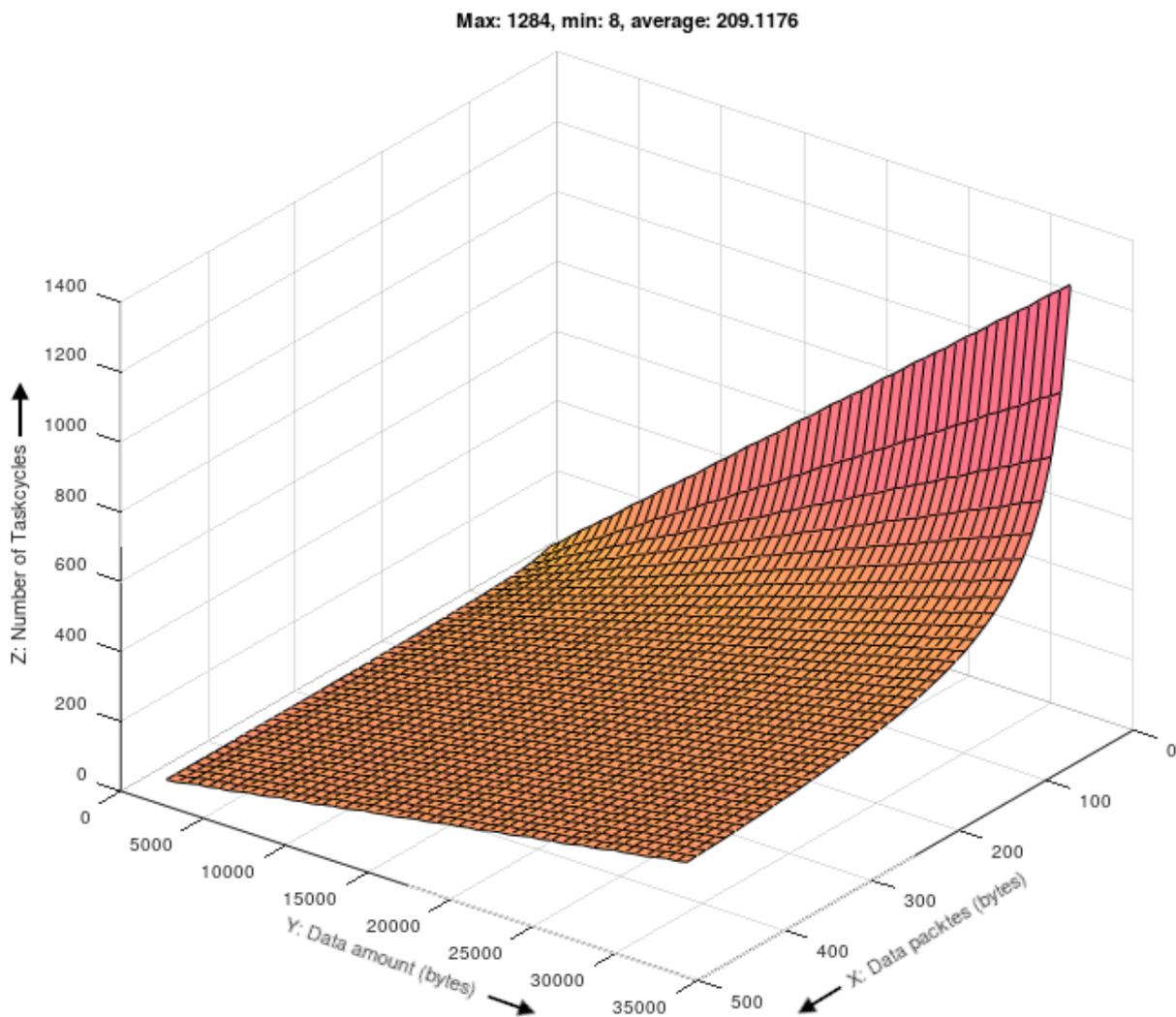


図 168: EL6695でのFoEスループットテスト: データパケットサイズおよびデータ量ごとのタスクサイクル数

ただし、これにはシーケンシャルデータアクセスによるシステム関連の実質値も含まれています(上記参照)。これらが記録され、各X/Yテスト点に対するZ軸の個々の値から減算された場合、結果はデータパケットサイズに依存しない準線形図曲線となります。

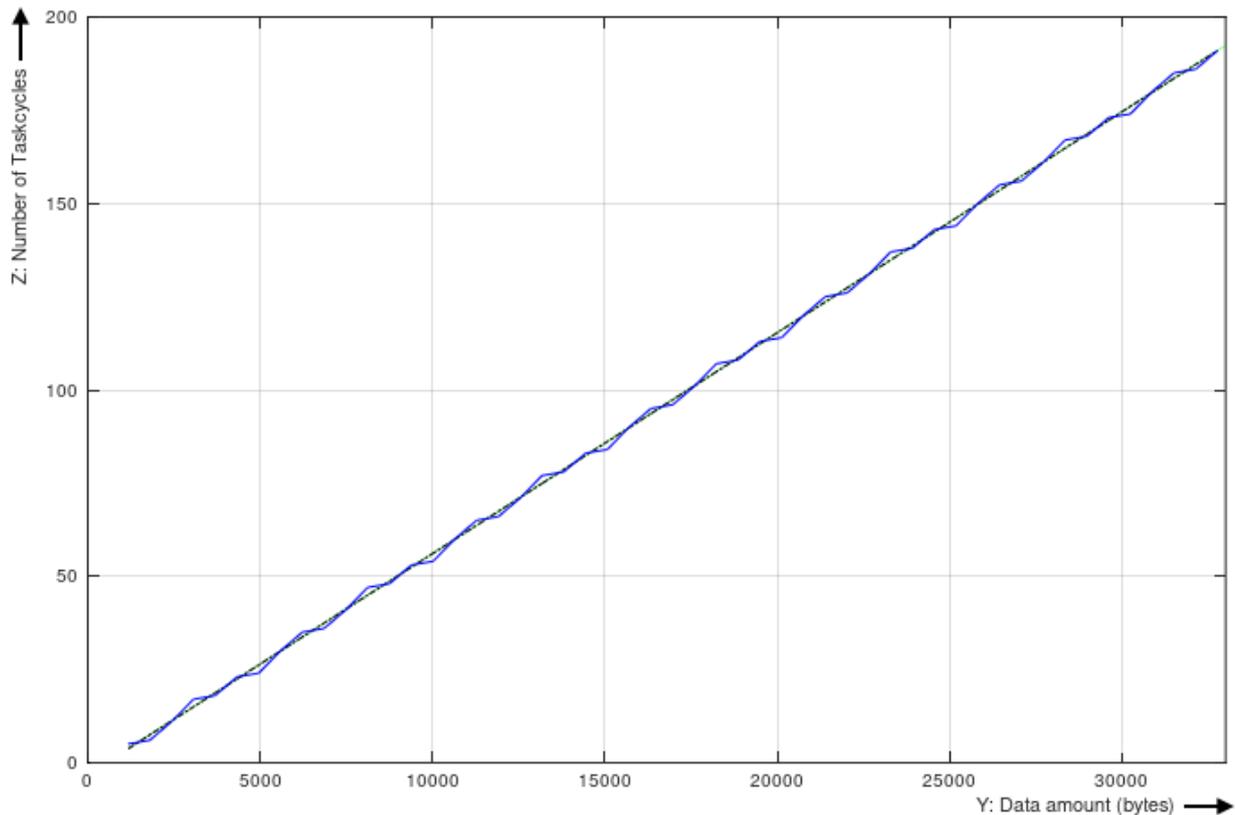


図 169: EL6695のFoEデータスループットの例について

線形近似では、このFoEの例でのスループットは約171 kB/sとなります。

6.6.5 VoE – vendor-specific protocol over EtherCAT

VoEによってユーザ固有のプロトコルの実装が可能になるため、メールボックス通信で新しく定義した特殊なデータ送信を実現できます。

詳細情報については、関連データ (TwinCATバージョン、必要なパフォーマンス値、サイクルタイム、データ量など) をご用意の上、ベッコフサポートまでお問い合わせください。

6.6.6 SoE – Servo Drive Profile over EtherCAT

詳細情報については、関連データ (TwinCATバージョン、必要なパフォーマンス値、サイクルタイム、データ量など) をご用意の上、ベッコフサポートまでお問い合わせください。

6.7 ディストリビュートクロック

● EtherCATマスタでのDCサポート



EL6695のディストリビュートクロックユニットには、DCシステム時間が用意されていますが、同期/ラッチユニットはありません。対応する初期化ルーチンは、TwinCAT 3のb4018.4以降、またはTwinCAT 2のb2248以降でサポートしています。

EL6695のコンフィグレーションは、タブ内で確認できます。データ交換のデフォルト設定は、「同期なし」(「FreeRun」)です。

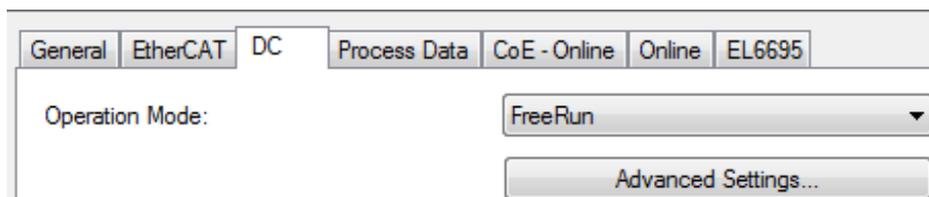


図 170: ディストリビュートクロック: 同期なし

EL6695には完全な2つのEtherCATスレーブが搭載されているため、その2つのディストリビュートクロックユニットは基本的に互いに独立しています。EL6695は2つのDC同期メカニズムをサポートしています。

- ・ 前のモデル (EL6692) と同様、EL6695は同期スレーブ側で使用可能な内部/外部タイムスタンプ情報を作成できるため、EL6695内のEtherCATマスタは自身のリアルタイム/EtherCATサイクルを調整できます。双方向の制御が可能です。
 - TwinCATでのアプリケーション: EL6695をDCモードに設定することで、TwinCATはEL6695をリファレンスクロックとして使用できます。

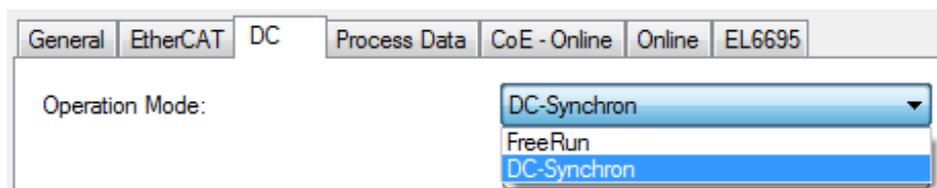


図 171: ディストリビュートクロック: 同期あり

- 32または64ビットタイムスタンプPDO 0x1a02または0x1a03を表示し、有効にします。これにより、TwinCATはターミナルを外部リファレンスクロックとして認識して、タイムスタンプを読み込むようになります。

Sync Manager:

SM	Size	Type	Flags
0	1024	MbxOut	
1	1024	MbxIn	
2	0	Outputs	
3	22	Inputs	
4	128		

PDO Zuordnung (0x1C13):

- 0x1A01 (excluded by 0x1A02)
- 0x1A02
- 0x1A03 (excluded by 0x1A02)
- 0x1A08

図 172: タイムスタンプPDO 0x1A02または0x1A03の表示

- 外部リファレンスクロックは、[EtherCAT DC master settings]で選択できます。

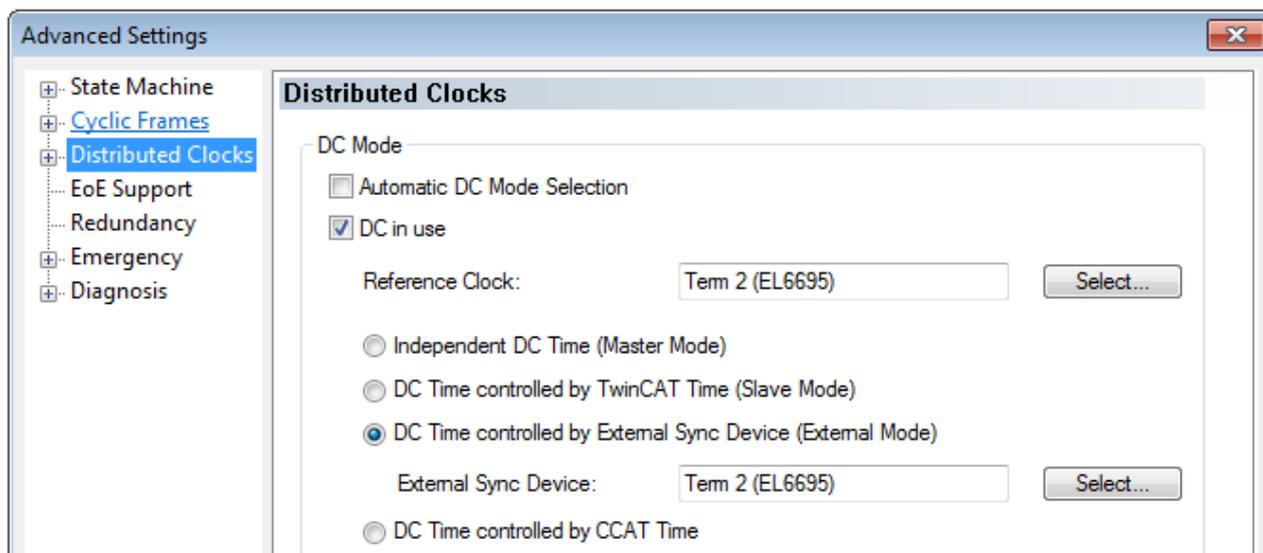


図 173: ディストリビュートクロック: 外部リファレンスクロックの選択(この例ではプライマリ)

この設定は、「SyncSlave」側でのみ必要です。

「SyncMaster」側では、EL3104 (DC対応)などを入力できます。

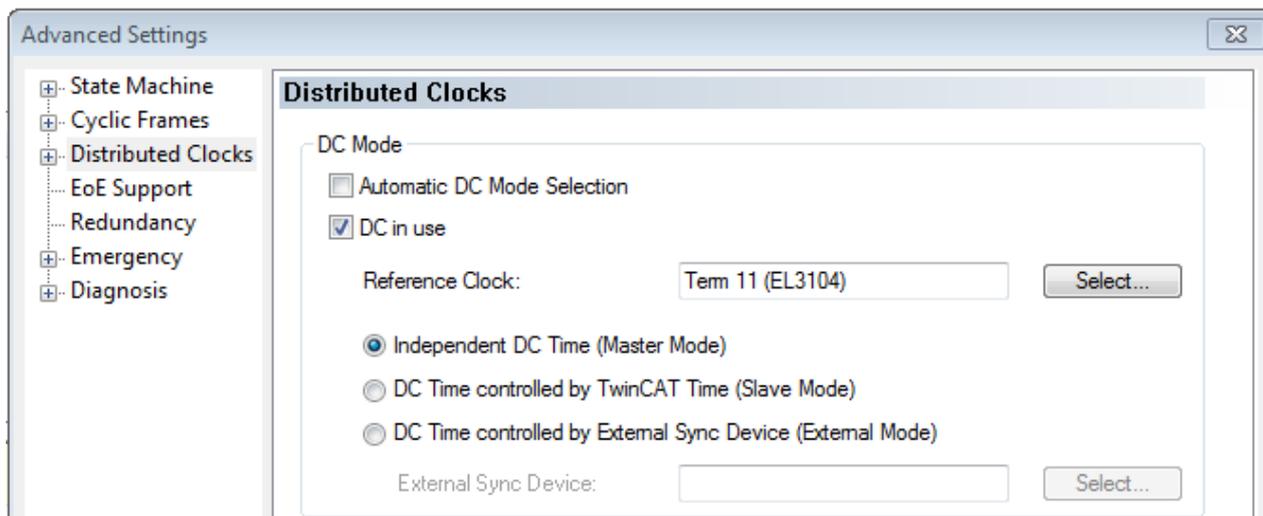


図 174: ディストリビュートクロック: 「SyncMaster」としてのEL3104の選択(この例ではセカンダリ)

- ・ EL6695が両方のEtherCATシステムで最初のDC対応スレーブである場合、EtherCATマスタは意識することなく2つのブリッジ側の直接DC結合を有効にできます。この場合、両方のEtherCATシステム/マスタがEL6695の時間にしががいます。特殊な設定は不要です。ただし、2つのシステム時間の間に一定のずれが生じます。
 - 双方のEtherCATマスタは、自身の0x0920 DCシステムタイムのずれを設定します。EL6695はこれを承認し、ESCとEL6695の同期性が確保されます。

● ローカル制御方式を未サポート

I EL6692とは異なり、EL6695はローカル制御方式「ControlValue for DC MasterClock」をサポートしていません。EtherCATマスタは、前述のタイムスタンプ方式を実装する必要があります。

EL6695はディストリビュートクロックの時間情報を内部的に使用せず、他方への転送しか行わないため、ESC内における特殊なDCレジスタの使用は不要であり、サポートしていません。DCをサポートする多くのEtherCATデバイスでは、PreOpからSafeOpへのステータス遷移中に、マスタがDCレジスタ0x09A0、0x0990、0x0980、0x09A8を書き込みます。EL6695にはこれらのレジスタがないため、スタートアップ中に書き込みを試行しても、EtherCATマスタから「Init Cmd failed: set DC cycle time」などのエラーメッセージが出力されません。

IP	PS	PI	SP	SO	SI	OS	OP	OI	IB	BI	CMD	Comment
	X										FPWR	set DC cycle time
	X										FPWR	set DC start time
	X										FPWR	set DC activation
	X										FPWR	set DC latch0 cfg

図 175: ディストリビュートクロック: エラーメッセージ

このため、EL6695ではEtherCATマスタがこれらのレジスタの書き込み、または書き込みコマンドのWcStateの評価を回避する必要があります。EL6695-ESIはこのDC機能をフラグ`TimeLoopControlOnly = TRUE`と記述し、EtherCATマスタはこれを認識できます。

6.8 オンラインスキャン

EL6695は、FoEによって書き込みおよび読み取り可能なあらゆるファイルに対して、プライマリとセカンダリでそれぞれ128kbyteのメモリを使用できます。

このメモリは、マスタが完全なESI (EtherCAT slave information)を受信して、そのESIを自身のコンフィギュレーションで使用できるようにするための、エミュレーションESIの保存に使用されます。

補足: 例

6.9 EL6692互換モード

EL6695には、いわゆるEL6692互換モードが用意されています。

これらのモードでは、従来のEL6692のように動作します。これにより、高性能なEL6695をEL6692インターフェイス用に設計されたEtherCATマスタおよびアプリケーションソフトウェアで動作させることが可能です。これらのモードでは、最初に同一のデフォルトプロセスデータが提供されます。

6.9.1 PDO MappingとAssignによるモジュール動作

このモジュール動作は基本的に標準モードであり、新しいEL6695用です。セクション「対称PDO Mapping」に説明が記載されています。

6.10 EL6695パフォーマンスモード

6.10.1 PDO Mappingの基本原則

廃止

これについて参照する

■ 対称PDO Mapping [▶ 111]

6.10.2 Initコマンドなし

主にこのモードは、プロセスデータ通信にのみ使用する外部マスタ用です。[Process Data]タブ内の自動PDO割り当ておよびマッピングのチェックボックスは、チェックしてはいけません(セクション「選択的PDO Mapping [▶ 115]」を参照)。

これにより、ターミナルのプライマリおよびセカンダリで変数を宣言できます。例えば、3バイトの出力変数が宣言されている場合、デバイスを再起動する(TwinCAT: [Reload Devices])と、PD出力オブジェクトの構造を各CoEオブジェクト0x1608内に表示できます。PD出力オブジェクトの長さは24ビットで、3バイトの変数があります。

1608.0	IO RxPDO-Map 8	RW	> 1 <
1608.01	SubIndex 001	RW	0x7000:01, 24

図 176: 24ビットのCoEオブジェクト0x1608

ターミナルの他方では、データ交換する場合にPDO AssignおよびPDO Mappingを無効にする必要があります。これは、ターミナルがデータを解釈し、それぞれ8ビットの3つのエントリではなく、24ビットの1つのエントリを想定できるようにするためです。さらに、リモート側のプロセスデータ長が正確に一致している必要があります。この例では、CoEオブジェクト0x1A01のマッピングをリモート側で無効にする必要があります(セクション「対称PDO Mapping [▶111]」を参照)。無効にしないと、異なる入力データ量が想定され、データ交換ができなくなります。

6.10.3 オブジェクトディスクリプションのダウンロード

EtherCAT通信デバイスは、オブジェクトの概要および内部機能や通信のパラメータとしてのCoEディレクトリを用意しています。EL6695ブリッジターミナルには、初期状態で対応するオブジェクトディレクトリ(OD)が用意されており、これがEL6695に適用されます。

オブジェクトディレクトリ(OD)を全体的または部分的に変更すると、EL6695ターミナル上にユーザ固有のデバイスを定義できます。TwinCAT System Managerのエクステンションにより、必要に応じて全体的または部分的なODを作成でき、ベッコフの内部ファイル形式である*.coeファイルとして保存できます。

6.10.4 デバイスエミュレーション

デバイスエミュレーション時は、ブリッジ側ではブリッジターミナル以外のEtherCATスレーブはエミュレートされます。物理的にはEL6695ブリッジターミナルはEtherCATスレーブであり、必ずこの元の状態にリセットされます。

選択したEL6695にESIファイルを使用し、エミュレートするEtherCATスレーブの情報をロードします。EL6695はこのESIデータを引き継いで、識別データ、PDO、およびCoEの形式でスレーブをエミュレートします。ただし、これは形式的な動作のみであり、EL6695のファームウェアに変更が必要となるような動作は含まれません。ブリッジ先の側でエミュレートするスレーブの「動作」をあらかじめPDOデータをマッピングし、そのスレーブの動作のようにPDOデータを供給する必要があります。

この方法で表現されたスレーブはEmulationModeでのEtherCAT ETGコンFORMANCEステータスに合格する必要があるため、ユーザはEmulationModeでEL6695に有効/認証済み/認証可能なESIのみロードするする必要があります。このため、ETG仕様書 ETG1000.6 (Mapping)、およびETG2000 (ESI仕様)に準拠する必要があります。

EmulationModeモードは、片側でも両側でも使用可能です。

● エクステンション(TwinCATのユーザインターフェイス)

i いわゆるエクステンション(TwinCATのユーザインターフェイス)は、エミュレートするスレーブでは使用できません。

詳細情報については、関連データ(TwinCATバージョン、必要なパフォーマンス値、サイクルタイム、データ量など)をご用意の上、ベッコフサポートまでお問い合わせください。

6.11 アプリケーション固有の変数定義

EL6695はデフォルト設定でプロセスデータを送受信しないため、アプリケーション要件にあうようにコンフィグレータで設定する必要があります。使用しているEtherCATマスタが変数のPDO Mappingをサポートしていない場合、ユーザはそのコンフィグレータで使用するESIデバイス記述ファイルを独自に作成できます。

● 技術情報

i 以下の取扱説明書のとおり、ETG仕様書 ETG1000.6 (Mapping)、およびETG2000 (ESI仕様)に準拠する必要があります。

● 法令

i デバイス記述ファイルがETGのコンFORMANCEステータスに合格する必要があります。各ユーザ/作成者も作成する記述ファイルのETGによる認証に注意が必要です。

7 付録

7.1 EtherCAT ALステータスコード

詳細情報は、[EtherCATシステムの説明](#)を参照してください。

7.2 ファームウェアの互換性

ベッコフEtherCATデバイスは、利用可能な最新のファームウェアバージョンが搭載された状態で提供されま
す。ファームウェアとハードウェアとの互換性は必要条件です。すべての組合せに互換性があるわけではあ
りません。以下の概要に、ファームウェアバージョンに対し動作可能なハードウェアバージョンを記載して
います。

ヒント

- ・ 各ハードウェアに対して使用可能な最新のファームウェアを使用することを推奨します。
- ・ ベッコフは納品済みの製品に対してお客様に無償でファームウェア更新を提供する義務はありません。

注記

デバイス損傷のリスク

該当ページ [▶ 150]に記載されているファームウェア更新の方法をご確認ください。

ファームウェア更新のためデバイスをBOOTSTRAPモードに移行し、ダウンロードを行っている間には新しい
ファームウェアが適切かどうかはチェックされません。
これにより、デバイスが損傷する可能性があります。
のため、ハードウェアバージョンに対してファームウェアが適切かどうかを操作前に必ずご確認ください。

EL6695

ハードウェア (HW)	ファームウェア (FW)	リビジョン番号	リリース日付
02 - 05*	03	プライマリ: EL6695-0000-0001 セカンダリ: EL6695-0002-0001	2015/02
		04	プライマリ: EL6695-0000-0002 セカンダリ: EL6695-0002-0002
	05	プライマリ: EL6695-0000-0003 セカンダリ: EL6695-0002-0003	2015/06
		06	プライマリ: EL6695-0000-0004 セカンダリ: EL6695-0002-0004
	07	プライマリ: EL6695-0000-0005 セカンダリ: EL6695-0002-0005	2017/10
	08*		2018/04

*) これは、本取扱説明書作成時において、互換性のある最新のファームウェア/ハードウェアバージョンで
す。新しい取扱説明書がないか、ベッコフのWebページをチェックしてください。

7.3 ファームウェア更新EL/ES/EM/ELM/EPxxxx

このセクションでは、ベッコフEtherCATスレーブEL/ES、ELM、EM、EK、およびEPシリーズのデバイス更新について説明します。ファームウェアの更新は、必ずベッコフサポートにご相談の上、行ってください。

ストレージの場所

EtherCATスレーブは、動作データを最大で3か所に保存します。

- ・ EtherCATスレーブは機能および性能によって、1つまたは複数のI/Oデータ処理用ローカルコントローラを搭載しています。対応するプログラムは、*.efw形式のいわゆる**ファームウェア**です。
- ・ EtherCATスレーブによっては、EtherCAT通信もこれらのコントローラに追加されています。この場合、このコントローラは通常、*.rbfファームウェアを使用するいわゆる**FPGAチップ**です。
- ・ 加えて、EtherCATスレーブは自身のデバイス記述ファイル(ESI: EtherCAT Slave Information)を保存するためのメモリチップである、いわゆる**ESI-EEPROM**を搭載しています。電源投入時、この記述ファイルがロードされ、それに応じてEtherCAT通信がセットアップされます。デバイス記述ファイルは、ベッコフウェブサイト (<https://www.beckhoff.de>) のダウンロードページから入手できます。ここでは、すべてのESIファイルをzipファイルとして取得できます。

お客様は、EtherCATフィールドバス、およびその通信メカニズムを使用してデータにアクセスできます。これらのデータの更新や読み取りには、非同期メールボックス通信、またはESCへのレジスタアクセスが使用されます。

スレーブがこの用途でセットアップされている場合、TwinCAT System Managerは3つのパートをすべて新しいデータでプログラミングするメカニズムを提供します。通常、スレーブは新しいデータが適しているかをチェックしないため、データが適していない場合はスレーブが動作できなくなります。

バンドルファームウェアによる簡単な更新

いわゆる**バンドルファームウェア**を使用すると、更新がより簡単に行えます。この場合、コントローラのファームウェアとESIが*.efwファイル内で結合されます。更新中に、ファームウェアとESIの両方がターミナル内で変更されます。これを行うには、以下が必要となります。

- ・ 結合形式にするファームウェアは、ファイル名で認識できるようにし、「ELxxxx-xxxx_REV0016_SW01.efw」のようにリビジョン番号も含んでいること。
- ・ ダウンロードダイアログにパスワードとして「1」を入力すること。パスワードが「0」（デフォルト設定）の場合は、ESIは更新されず、ファームウェアの更新のみが実行されます。
- ・ この機能をサポートするデバイスにおいて、通常、この機能は変更できません。この機能は2016年以降に新規開発された多くの機能を含むコンポーネントです。

更新後、正常に更新されたかどうかを確認します。

- ・ ESI/リビジョン: TwinCAT ConfigMode/FreeRunでオンラインスキャンを使用。この方法で、リビジョンを簡単に判定できます。
- ・ ファームウェア: デバイスのオンラインCoEを確認。

注記

デバイスの損傷のリスク

新しいデバイスファイルのダウンロード時には、以下に注意してください。

- ・ EtherCATデバイスへのファームウェアダウンロードが中断されてはいけません。
- ・ EtherCAT通信中にデータ欠損が発生してはいけません。CRCエラーやLostFramesを回避する必要があります。
- ・ 十分な電源を確保する必要があります。指定された信号レベルである必要があります。

更新プロセス中に誤作動が発生した場合は、EtherCATデバイスが使用できなくなり、メーカーによる再コミッショニングが必要となる可能性があります。

7.3.1 デバイスESIファイル/XML

注記

ESI/EEPROMの更新に関する注意

スレーブによっては、製造時の較正データおよびコンフィグレーションデータがEEPROM内に保存されています。これらのデータは更新中に上書きされ、復元できなくなります。

ESIデバイス記述ファイルは、スレーブにローカルに保存されており、スタートアップ時にロードされます。各デバイス記述ファイルには、スレーブ名(9つの文字/数字)およびリビジョン番号(4つの数字)から成る固有の識別子が付けられています。System Managerで構成された各スレーブの識別子は、[EtherCAT]タブ内に表示されます。

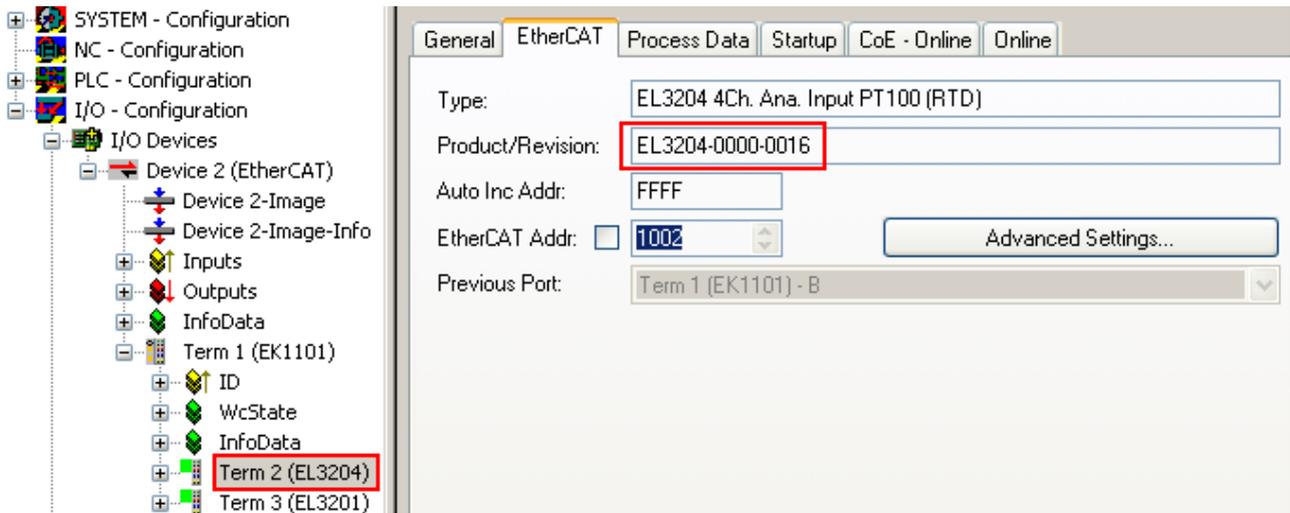


図 177: 名前「EL3204-0000」およびリビジョン「-0016」から成るデバイス識別子

設定されている識別子は、ハードウェアとして使用されている実際のデバイス識別子、つまりスレーブがスタートアップ時にロードした識別子(ここではEL3204)と互換性がある必要があります。通常、設定されているリビジョンは、ターミナルネットワーク内に実際に存在するリビジョン以下である必要があります。

これに関する詳細情報は、[EtherCATシステムの説明](#)を参照してください。

XML/ESIの更新

デバイスリビジョンは、使用するファームウェアおよびハードウェアと密接にリンクしています。組み合わせに互換性がないと、デバイスの誤作動やシャットダウンが発生します。対応する更新は、必ずベッコフサポートにご相談の上、行ってください。

ESIスレーブ識別子の表示

構成されているデバイス設定と実際のESIの内容との互換性を確認する最も簡単な方法は、TwinCATモードConfig/FreeRunでEtherCATボックスの検索です。

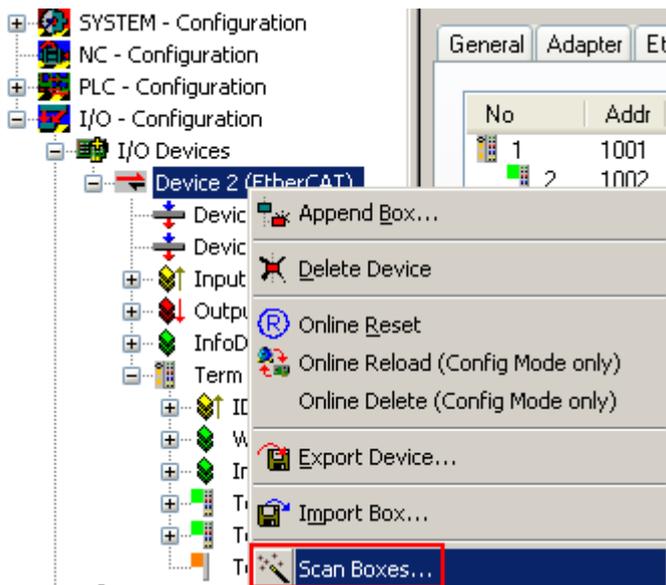


図 178: EtherCATデバイスを右クリックして下層のフィールドデバイスをスキャン

検出されたフィールドデバイスと構成されたフィールドデバイスが一致する場合は、以下が表示されます。



図 179: 設定が同一

フィールドデバイスが一致しない場合は、コンフィグレーション内に実際のデータを入力するための変更ダイアログが表示されます。

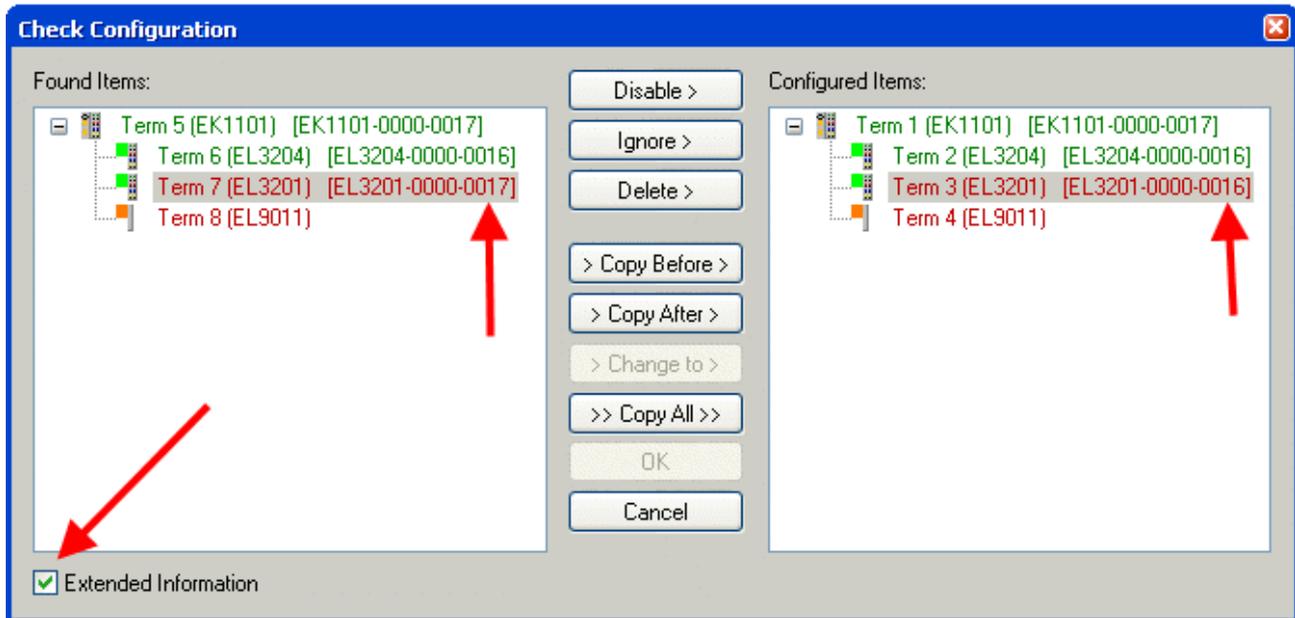


図 180: 変更ダイアログ

図. 「変更ダイアログ」の例では、EL3201-0000-0016が構成されているにも関わらず、EL3201-0000-0017が検出されています。この場合、[Copy Before]ボタンを使用してコンフィグレーションを適合できます。リビジョンを表示するには、[Extended Information]チェックボックスを設定する必要があります。

ESIスレーブ識別子の変更

ESI/EEPROM識別子は、TwinCATで以下のように更新できます。

- ・ スレーブと障害なくEtherCAT通信が確立されている必要があります。
- ・ スレーブの状態は関係ありません。
- ・ オンライン表示でスレーブを右クリックすると、[EEPROM Update]ダイアログが開きます(図. 「EEPROM Update」)

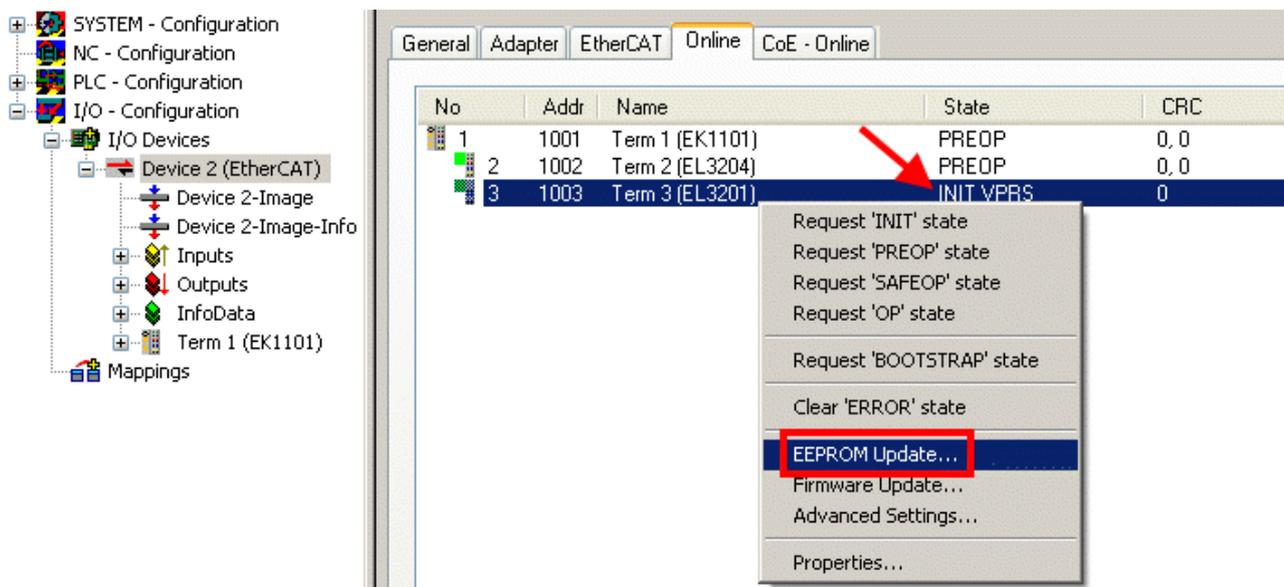


図 181: EEPROM Update

次のダイアログで、新しいESIを選択します(図。「新規ESIの選択」を参照)。「[Show Hidden Devices]」チェックボックスを有効にすると、通常は非表示のスレーブの旧バージョンも表示されます。

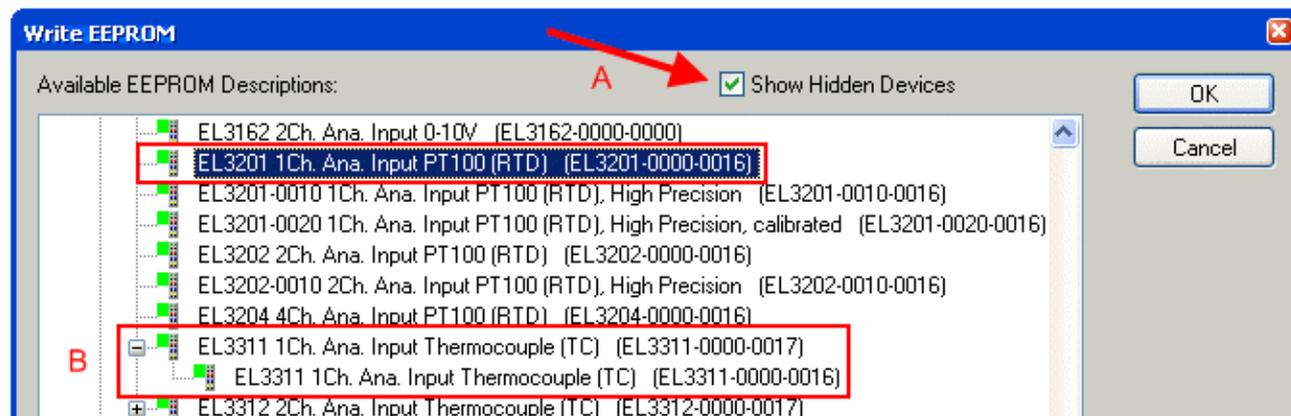


図 182: 新規ESIの選択

System Managerのプログレスバーに進捗が表示されます。データは書き込まれてから検証されます。

● 変更は再起動後に反映されます。

i ほとんどのEtherCATデバイスは変更したESIを直ちに、またはINITからのスタートアップ後に読み取ります。ディストリビュートクロックなどのいくつかの通信設定は、電源投入中にしか読み込まれません。このため、変更を反映するには、EtherCATスレーブのスイッチを短時間オフにする必要があります。

7.3.2 ファームウェアの説明

ファームウェアバージョンの判別

レーザ刻印されたバージョンの判別

ベッコフEtherCATスレーブには、シリアル番号がレーザで刻印されています。シリアル番号は、次のように構成されています: KK YY FF HH

- KK - 製造された週(CW、暦週)
- YY - 製造された年
- FF - ファームウェアバージョン
- HH - ハードウェアバージョン

シリアル番号の例: 12 10 03 02:

- 12 - 製造された週12
- 10 - 製造された年2010
- 03 - ファームウェアバージョン03
- 02 - ハードウェアバージョン02

System Managerによるバージョンの判別

マスタがスレーブにオンラインアクセスできる場合は、TwinCAT System Managerにはコントローラファームウェアのバージョンが表示されます。コントローラファームウェアをチェックするEバスターミナル(この例ではターミナル2 (EL3204))をクリックし、タブ[CoE Online] (CAN over EtherCAT)を選択します。

● CoEオンラインおよびオフラインCoE

i

2つのCoEディレクトリが用意されています:

- ・ **online**: EtherCATスレーブがこれをサポートしている場合は、このCoEディレクトリがコントローラによってEtherCATスレーブ内に提供されます。スレーブが接続されており、動作可能な状態である場合のみ、このCoEディレクトリを表示できます。
- ・ **offline**: EtherCATスレーブ情報ESI/XMLには、CoEのデフォルトのコンテンツを含めることが可能です。ESIにこれが含まれている場合(「Beckhoff EL5xxx.xml」など)のみ、このCoEディレクトリを表示できます。

2つのビューを切り替えるには、[Advanced]ボタンを使用します。

図. 「EL3204のファームウェアバージョンの表示」では、選択されているEL3204のファームウェアバージョンが、CoEエントリ0x100A内の03として表示されています。

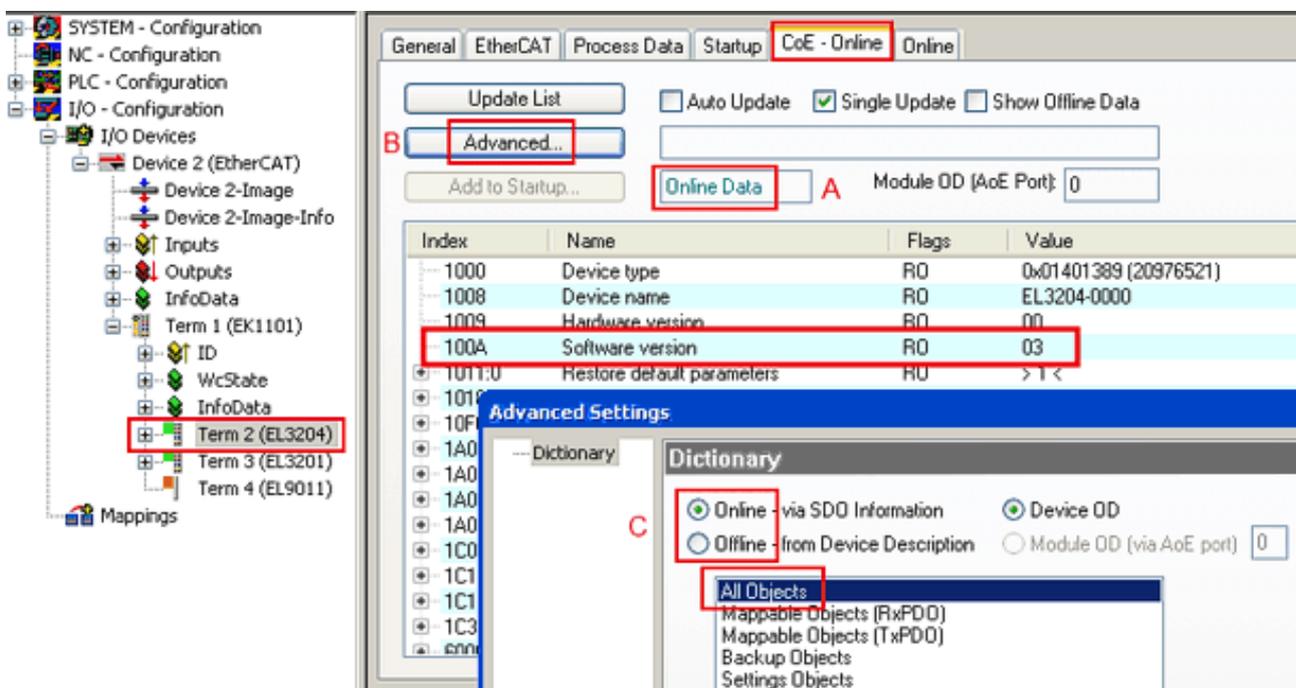


図 183: EL3204のファームウェアバージョンの表示

TwinCAT 2.11には、現在Online CoEディレクトリが表示されています(A)。表示されていない場合は、[Advanced]設定(B)の[Online]オプションで[All Objects]をダブルクリックするとOnlineディレクトリをロードできます。

7.3.3 コントローラファームウェア*.efwの更新

● CoEディレクトリ

i

Online CoEディレクトリはコントローラによって管理され、専用のEEPROM内に保存されます。通常、これはファームウェア更新中には変更できません。

コントローラのファームウェアを更新するには、[Online]タブに切り替えます(図. 「ファームウェア更新」)。

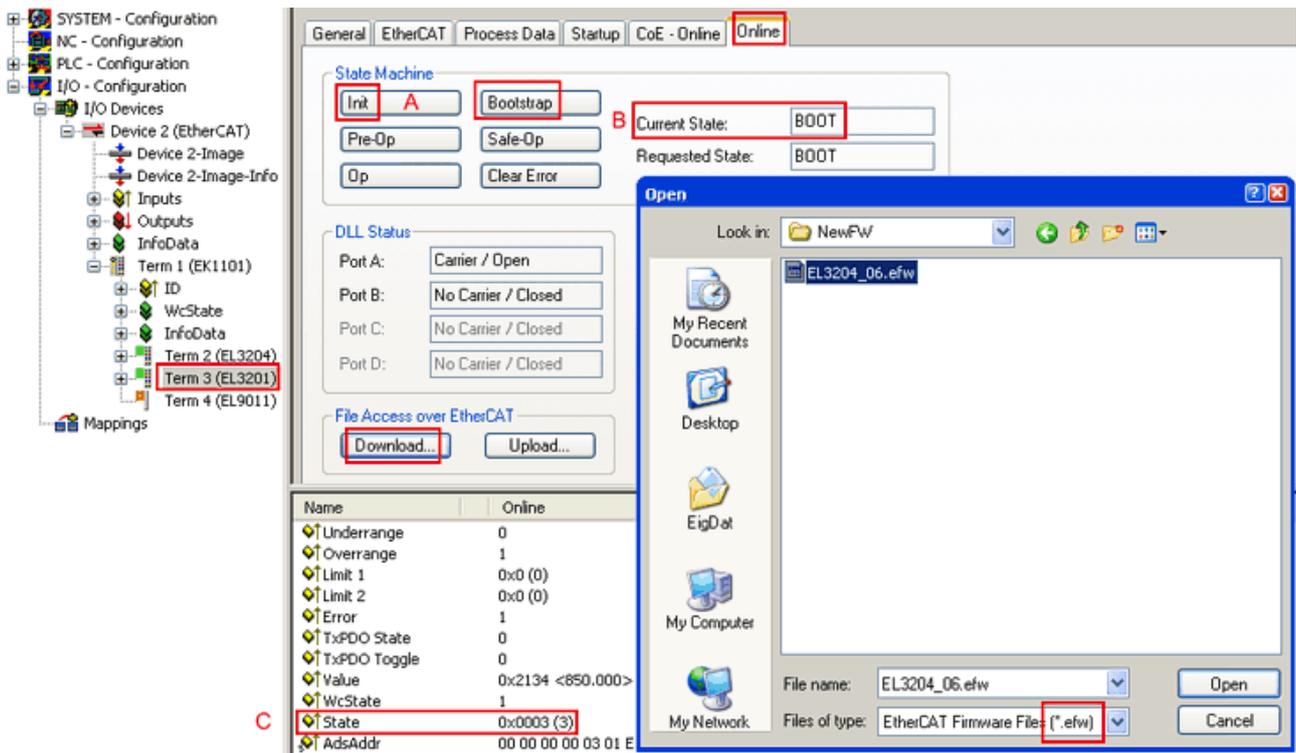
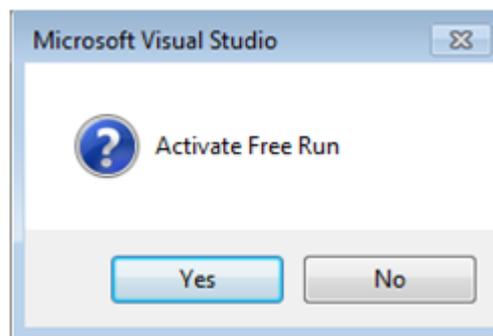
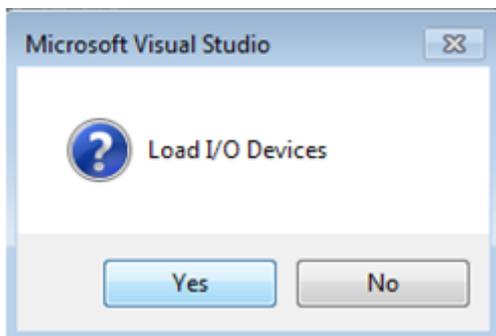


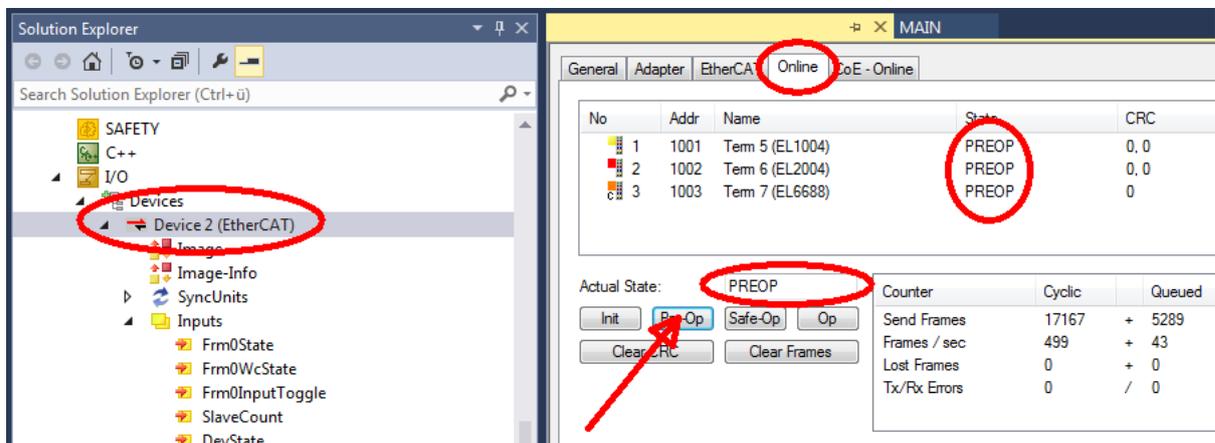
図 184: ファームウェア更新

ベッコフサポートの指示がない限り、以下の手順を実行します。EtherCATマスタとしてのTwinCAT 2および3で有効です。

- ・ TwinCATシステムをConfigMode/FreeRunに切り替え、サイクルタイムを1 ms以上に設定します (ConfigModeのデフォルトは4 ms)。リアルタイム動作中のファームウェア更新は推奨されません。

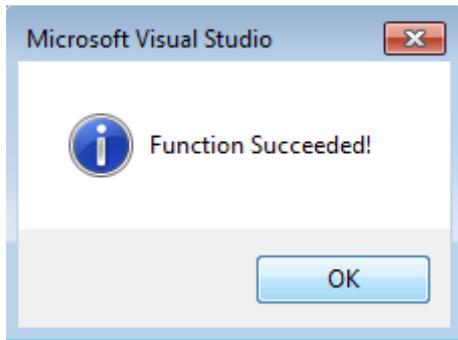


- ・ EtherCATマスタをPreOPに切り替えます。



- ・ スレーブをINITに切り替えます (A)。
- ・ スレーブをBOOTSTRAPに切り替えます。
- ・ 現在のステータス (B、C) をチェックします。

- ・ 新しい*efwファイルをダウンロードします(終了するまで待機してください)。通常、パスワードは不要です。



- ・ ダウンロード後、INIT → PreOPと切り替えます。
- ・ スレーブを短時間オフに切り替えます(電圧がかかった状態で取り外さないでください)。
- ・ ファームウェアのステータスが正常に変更されたかをCoE 0x100A内でチェックします。

7.3.4 FPGAファームウェア*.rbf

FPGAチップでEtherCAT通信を処理している場合、*.rbfファイルが更新を行う場合があります。

- ・ I/O処理用のコントローラファームウェア
- ・ EtherCAT通信用のFPGAファームウェア (FPGA搭載のターミナルのみ)

ターミナルのシリアル番号に含まれるファームウェアバージョン番号は、両方のファームウェアコンポーネントに含まれています。いずれかのファームウェアコンポーネントを変更すると、このバージョン番号が更新されます。

System Managerによるバージョンの判別

TwinCAT System Managerは、FPGAファームウェアバージョンを表示します。EtherCATネットワークのイーサネットカード(この例では「Device 2」)をクリックし、[Online]タブを選択します。

[Reg:0002]列に、個々のEtherCATデバイスのファームウェアバージョンが16進数および10進数で表示されます。

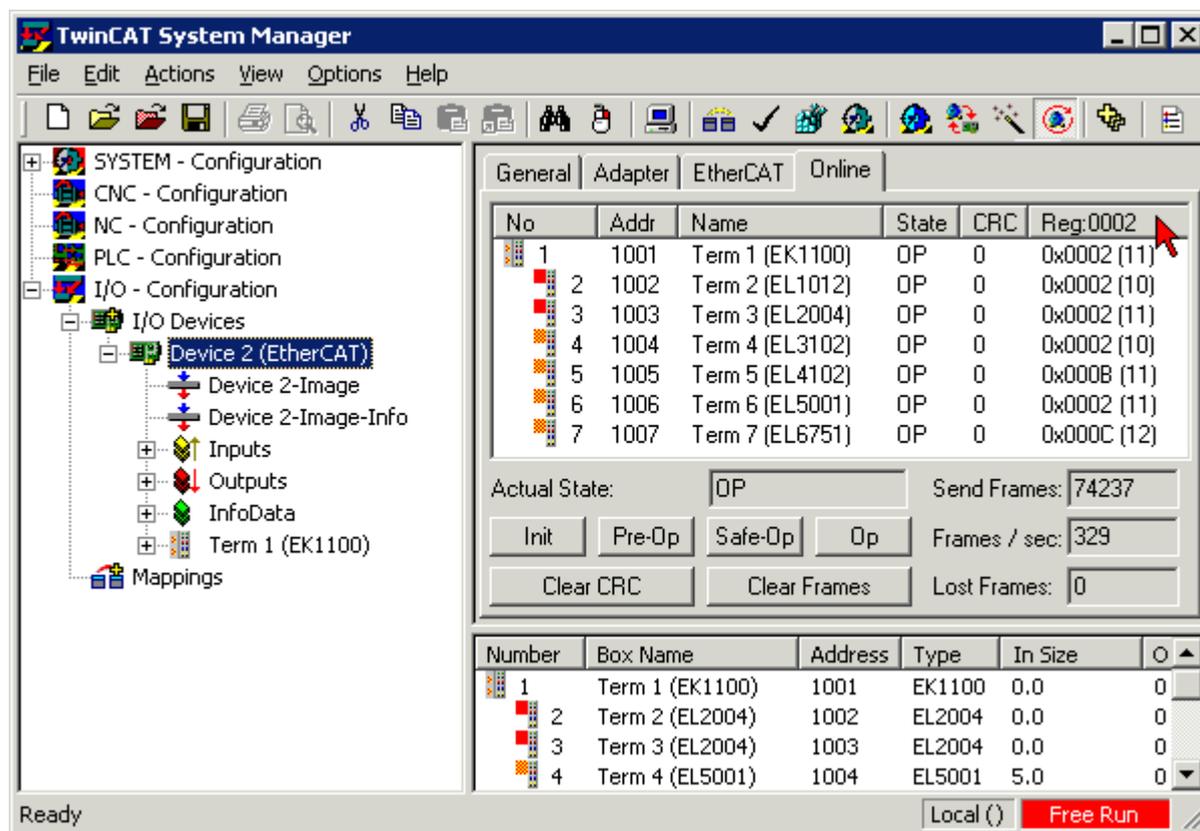


図 185: FPGAファームウェアバージョン定義

列[Reg:0002]が表示されていない場合は、テーブルヘッダを右クリックし、コンテキストメニューの[Properties]を選択します。

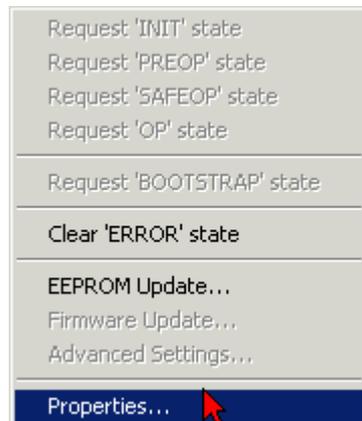


図 186: コンテキストメニュー[Properties]

表示される[Advanced Settings]ダイアログで、表示する列を選択できます。[Diagnosis → Online View]で、[0002 ETxxx Build]チェックボックスを選択してFPGAファームウェアバージョン表示を有効にします。

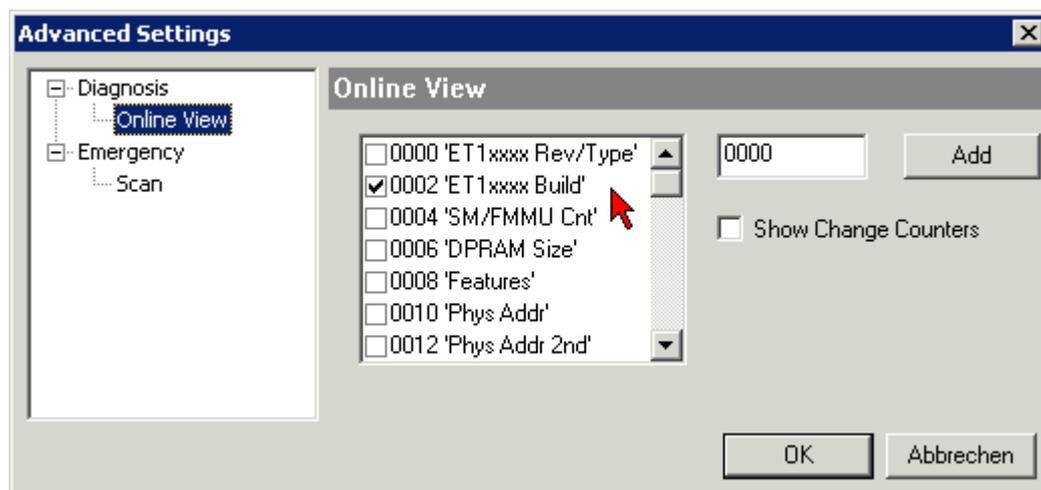


図 187: [Advanced settings] ダイアログ

更新

- ・ EtherCATカプラのFPGAファームウェアを更新する場合は、カプラのFPGAファームウェアバージョンが11以降である必要があります。
- ・ EバスターミナルのFPGAファームウェアを更新する場合は、ターミナルのFPGAファームウェアバージョンが10以降である必要があります。

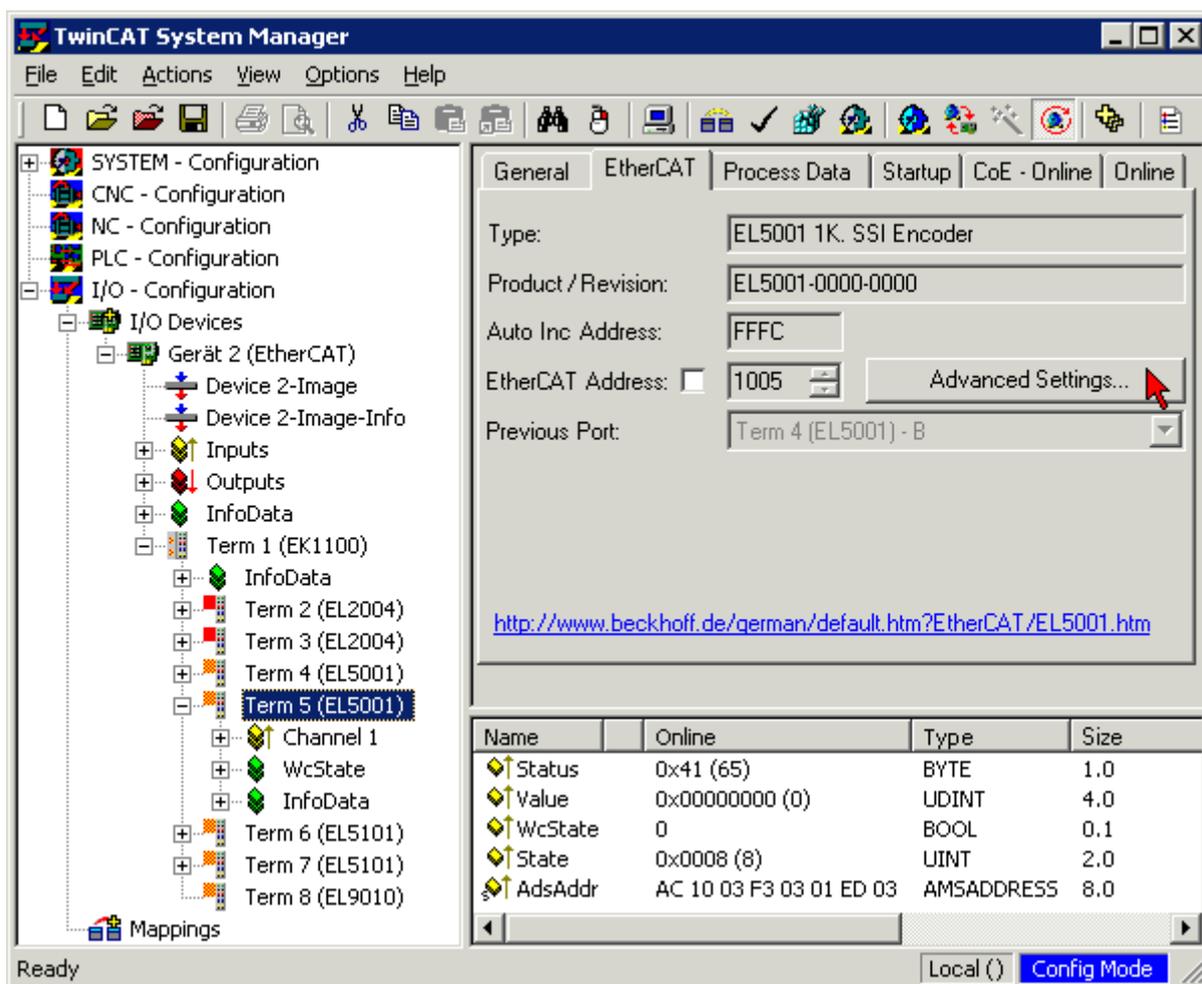
これ以前のファームウェアバージョンは、メーカーしか更新できません。

EtherCATデバイスの更新

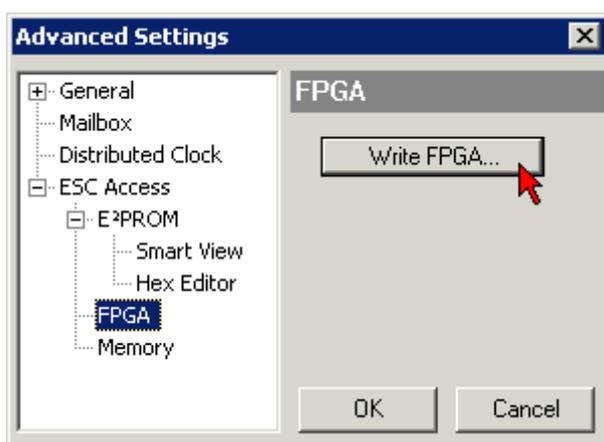
他に(ベッコフサポートなどによる)指定がない場合、以下の手順を遵守する必要があります。

- ・ TwinCATシステムをConfigMode/FreeRunに切り替え、サイクルタイムを1 ms以上に設定します (ConfigModeのデフォルトは4 ms)。リアルタイム動作中のファームウェア更新は推奨されません。

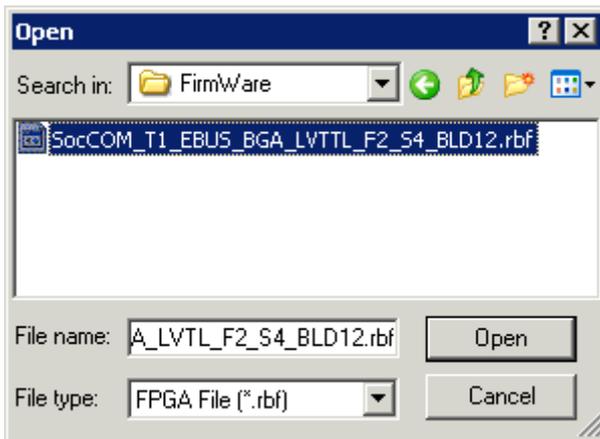
- ・ TwinCAT System Managerで、FPGAファームウェアを更新するターミナル(この例では「Terminal 5: EL5001」)を選択し、[EtherCAT]タブの[Advanced Settings]ボタンをクリックします。



- ・ [Advanced settings]ダイアログが表示されます。[ESC Access → E²PROM → FPGA]で、[Write FPGA]ボタンをクリックします。



- ・新しいFPGAファームウェアのファイル(*.rbf)を選択し、EtherCATデバイスに転送します。



- ・ダウンロードが完了するまで待機します。
- ・スレーブの電流を短時間オフにします(電圧がかかった状態で取り外さないでください)。新しいFPGAファームウェアを有効にするには、EtherCATデバイスを再起動する(電源をオフにし、再度オンにする)必要があります。
- ・新しいFPGAのステータスをチェックします。

注記

デバイスの損傷のリスク

ファームウェアのEtherCATデバイスへのダウンロードは、いかなる場合でも中断してはいけません。電源のスイッチオフや、イーサネットリンクの切断などでこの処理を中断すると、メーカーがEtherCATデバイスを再コミショニングできなくなります。

7.3.5 複数のEtherCATデバイスの同時更新

複数のデバイスのファームウェアファイル/ESIが同一である場合は、それらのデバイスのファームウェアおよびESIを同時に更新できます。

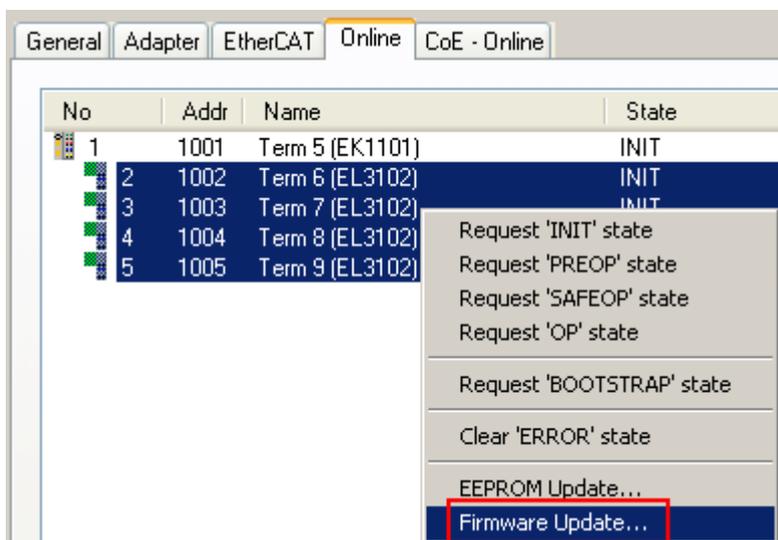


図 188: 複数選択およびファームウェア更新

目的のスレーブを選択し、前述のとおりBOOTSTRAPモードでファームウェア更新を実行します。

7.4 ファームウェア更新に関する補足

ファームウェア更新には2つの方法があります。1つは、[Online]タブのリストからデバイス(EL6695)を選択してから、ドロップダウンメニューで[Firmware Update...]を選択する方法です。これで、*.efwファイルに対応するパス経由でロードできます。パスワードは不要ですが、オプションで発行できます。更新後、ターミナルの両側がOP状態に復帰する前に、一度INIT状態に設定されます。

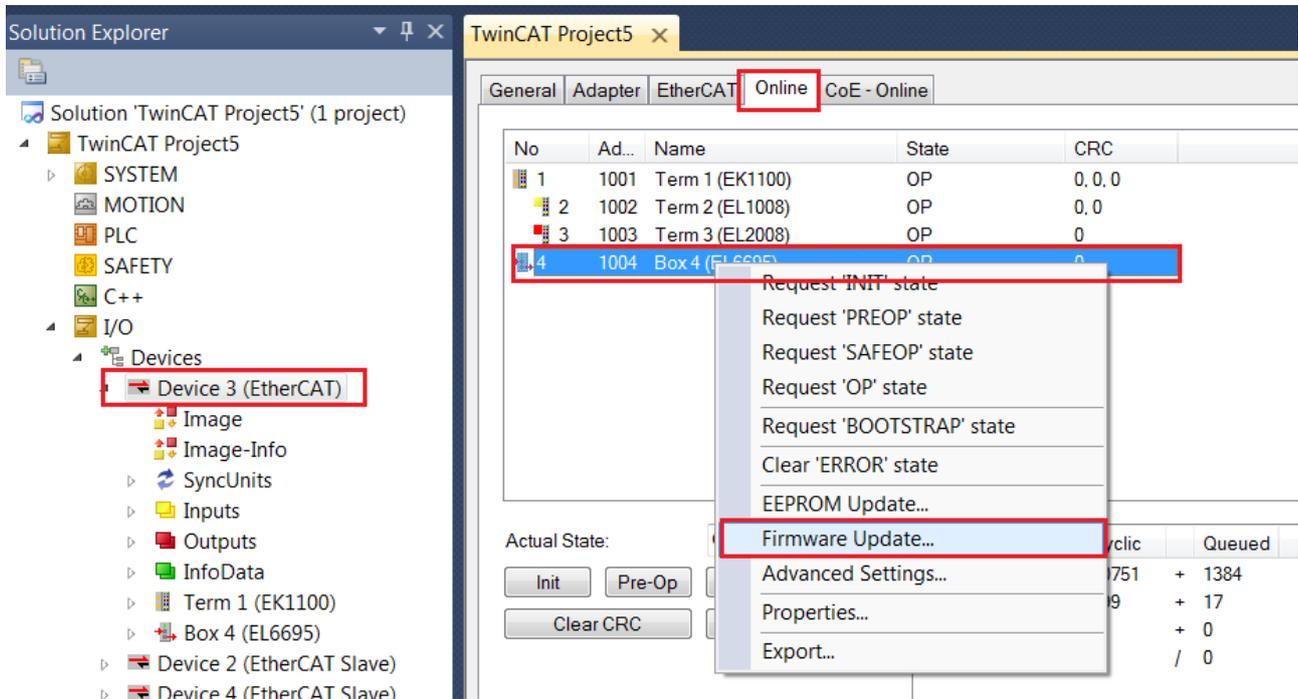


図 189: [Online]タブによるファームウェア更新

2つ目は、ターミナルおよび[Online]タブから直接行う方法です。これを行うには、ターミナルの選択した側をBOOT状態に設定します。

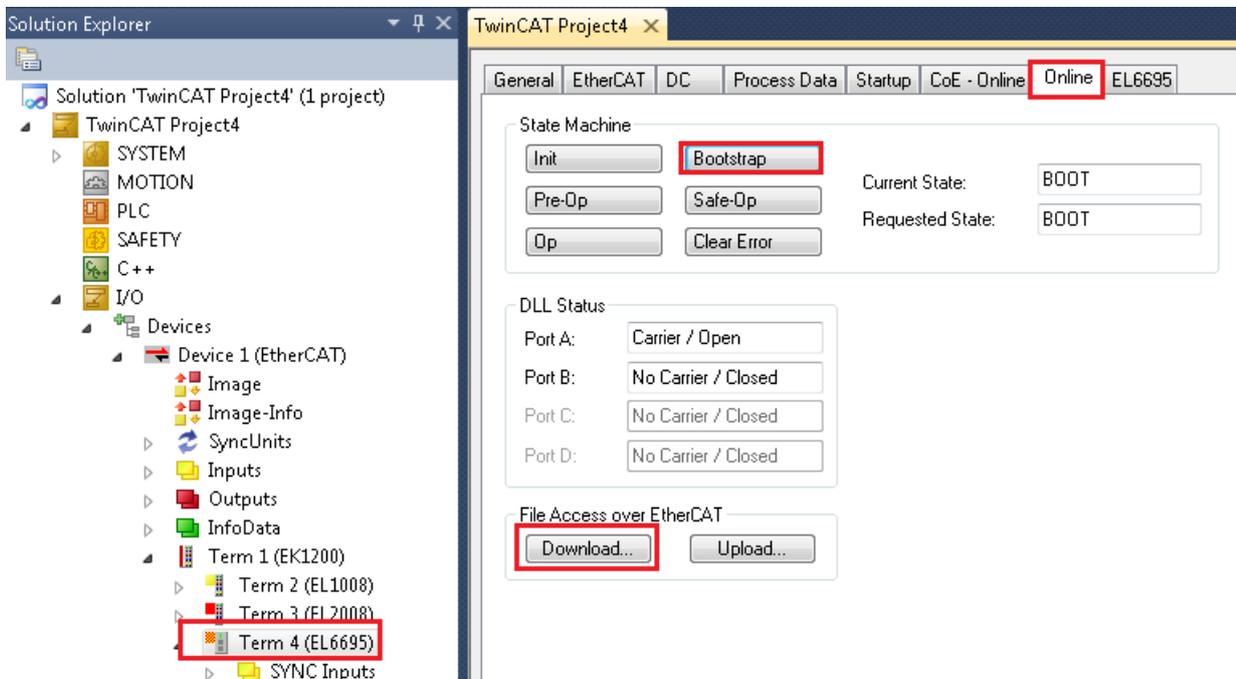


図 190: BOOT状態の設定

この状態で、[File Access over EtherCAT]の[Download]ボタンを使用して、*.efwファイルをターミナルにダウンロードできます。その後、ターミナルをINITに設定し、再度OPに設定します。逆側もINITに設定し、その後OPに設定する必要があります。

注記

FW更新(ベータFW)に関する注記

- 更新の前に、EtherCATバスがフレームやリンクの損失なく動作することを確認してください。
- システム内に複数のEL6695ターミナルが存在する場合、同時に複数のEL6695ターミナルに対してFWの更新を実行することはできません。同時に複数のEL6695ターミナルのFWを更新すると、デバイスが損傷して修復できなくなり、ベッコフのサービスによる交換が必要となる場合があります。
- 更新中は、電源障害が発生しないよう注意する必要があります。

7.5 診断

LED診断



図 191: LEDで動作状態を表示

LED	意味	
Run Prim Run Sec (対応するスレーブ側の動作状態)	消灯	EtherCATステートマシンの状態: INIT = ターミナルの初期化、または BOOTSTRAP = ターミナルのファームウェア更新用の機能
Run Prim Run Sec (対応するスレーブ側の動作状態)	緑の点滅	EtherCATステートマシンの状態: PREOP = メールボックス通信の状態、標準設定から変更
	緑が低速で点滅	EtherCATステートマシンの状態: SAFEOP = Sync Managerチャンネルおよびディストリビュートクロックの検証 出力が安全な状態
Power/Error (電源状態)	緑で点灯	EtherCATステートマシンの状態: OP = 通常の動作状態。メールボックスおよびプロセスデータ通信が可能
	消灯	供給なし
Power/Error (電源状態)	オレンジ	プライマリまたはセカンダリ的一方からのみ供給
	赤で点滅	FW更新に失敗、ベッコフのサービスにターミナルを送付
	緑	両側に供給中、24 V入力を使用

7.6 工場出荷状態の復元

ターミナルEL6695には、リセットの方法が2つあります。

- ・ コマンドを使用
- ・ CoEオブジェクト0x1011を使用 (FW07まで)

7.6.1 一般的なデバイスのリセット

EL6695内に保存されている設定は、コード0x33336695をCoEオブジェクト0xF008に入力するとリセットできます。

7.6.2 工場出荷状態の復元

ELxxxxターミナル内のバックアップオブジェクトを工場出荷状態に復元するには、CoEオブジェクトRestore default parameters、SubIndex 001をTwinCAT System Manager (Config Mode)で選択します(図。「Restore default parameters PDOの選択」を参照)。

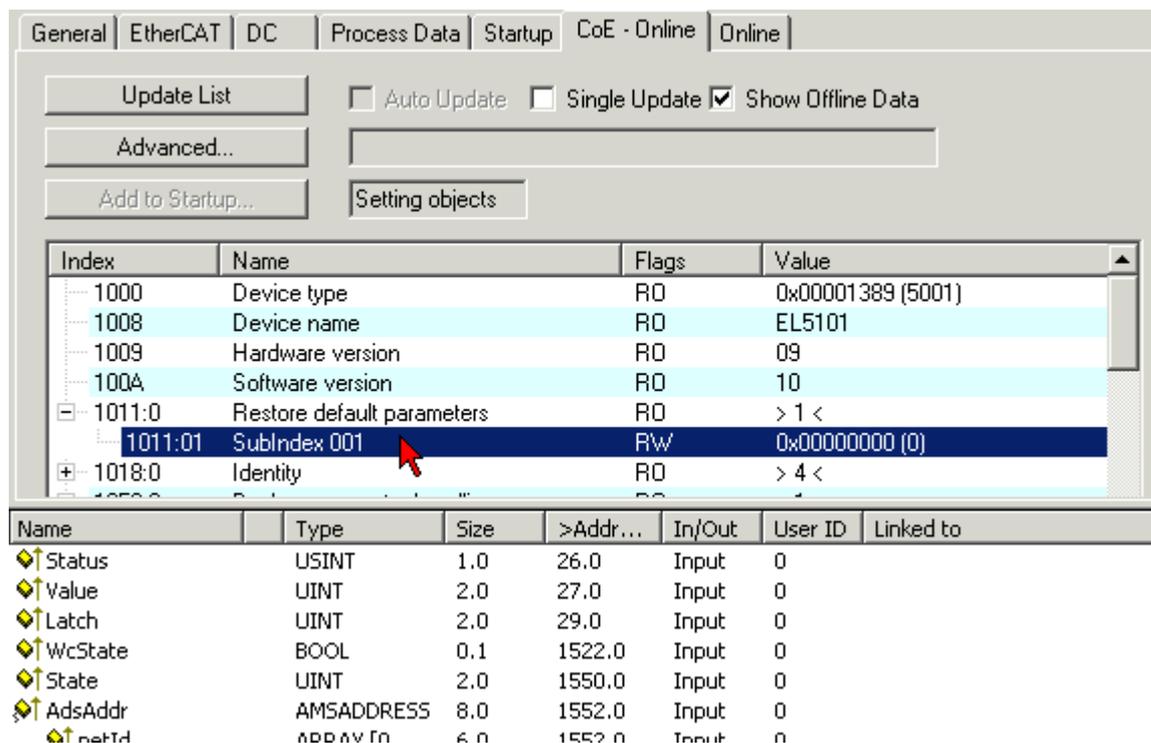


図 192: 「Restore default parameters」 PDOの選択

「SubIndex 001」をダブルクリックして、[Set Value]ダイアログを開きます。値1684107116をフィールド[Dec]に、または値0x64616F6Cをフィールド[Hex]に入力し、[OK]で確定します(図. 「[Set Value]ダイアログでの復元値の入力」)。すべてのバックアップオブジェクトが工場出荷状態にリセットされます。

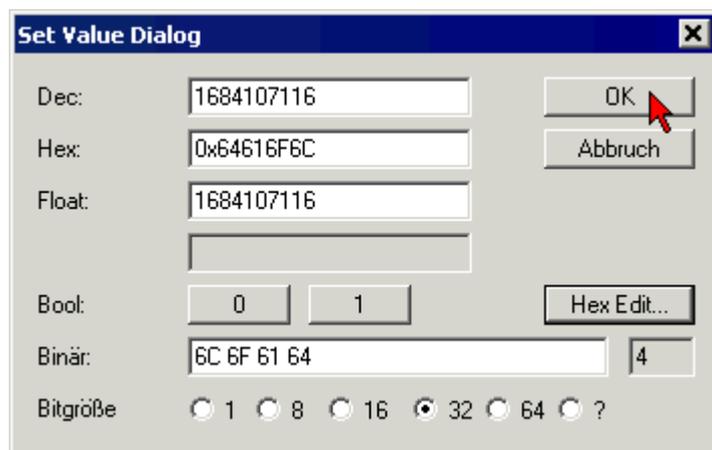


図 193: [Set Value]ダイアログでの復元値の入力

代替復元値

古いタイプのターミナルには、バックアップオブジェクトを代替復元値(10進数値: 1819238756、16進数値: 0x6C6F6164)で切り替えることができます。不正な復元値を入力しても動作しません。

7.7 サポートとサービス

世界中のベッコフ支社と代理店は、包括的なサポートとサービスを提供し、ベッコフ製品とシステムソリューションに関するあらゆる質問に対して迅速かつ的確なサポートを提供しています。

ベッコフの支社と代理店

ベッコフ製品に対するローカルサポートおよびサービスについては、最寄りのベッコフ支社または代理店にお問い合わせください。

世界中のベッコフ支社と代理店の所在はベッコフウェブ (<http://www.beckhoff.co.jp>) よりご確認ください。

また、このウェブページでベッコフ製品に関する取扱説明書も公開されています。

ベッコフ本社

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

電話: +49 5246 963 0
ファックス: +49 5246 963 198
電子メール: info@beckhoff.com

ベッコフサポート

ベッコフサポートはベッコフ製品に関するお問い合わせだけでなく、その他のあらゆる包括的な技術サポートを提供しています。

- ・ サポート
- ・ 複雑なオートメーションシステムの設計、プログラミングおよびコミショニング
- ・ ベッコフシステムコンポーネントに関する広範なトレーニングプログラム

ホットライン: +49 5246 963 157
ファックス: +49 5246 963 9157
電子メール: support@beckhoff.com

ベッコフのサービス

ベッコフサービスセンタは、すべてのアフターサービスでお客様をサポートいたします。

- ・ オンサイトサービス
- ・ 修理サービス
- ・ スペアパーツサービス
- ・ ホットラインサービス

ホットライン: +49 5246 963 460
ファックス: +49 5246 963 479
電子メール: service@beckhoff.com

図のリスト

図 1	シリアル/バッチ番号、およびリビジョンIDが記載されたEL5021 ELターミナル、標準IP20 I/Oデバイス(2014年1月以降の印字)	8
図 2	シリアル/バッチ番号が記載されたEK1100 EtherCATカプラ、標準IP20 I/Oデバイス	8
図 3	シリアル/バッチ番号が記載されたCU2016スイッチ	9
図 4	シリアル/バッチ番号26131006および固有のID番号204418が記載されたEL3202-0020	9
図 5	バッチ番号/日付コード22090101および固有のシリアル番号158102が記載されたEP1258-00001 IP67 EtherCATボックス	9
図 6	バッチ番号/日付コード071201FFおよび固有のシリアル番号00346070が記載されたEP1908-0002 IP67 EtherCAT安全ボックス	9
図 7	バッチ番号/日付コード50110302および固有のシリアル番号00331701が記載されたEL2904 IP20安全ターミナル	10
図 8	固有のID番号(QRコード) 100001051およびシリアル/バッチ番号44160201が記載されたELM3604-0002ターミナル	10
図 9	Data Matrixコードで表す BIC(DMC、コードスキームECC200)	11
図 10	EL6695	13
図 11	System Managerでの電流計算	16
図 12	[EtherCAT]タブ → [Advanced Settings] → [Behavior] → [Watchdog]	18
図 13	EtherCATステートマシンの状態	19
図 14	[CoE Online]タブ	21
図 15	TwinCAT System Managerに表示されたスタートアップリスト	22
図 16	オフラインリスト	23
図 17	オンラインリスト	23
図 18	ベッコフI/O機器のデータ通信用端子	26
図 19	標準設置方向の推奨距離	30
図 20	その他の設置方向	31
図 21	正しい配置	32
図 22	間違った配置	32
図 23	EL6695の接続	34
図 24	ユーザ側(コミッショニング)と設置の関係	35
図 25	組込み型PCを使用した制御コンフィグレーション、入力(EL1004)および出力(EL2008)	36
図 26	デフォルトのTwinCAT 2ユーザインターフェイス	37
図 27	ターゲットシステムの選択	38
図 28	アクセスするPLCをTwinCAT System Managerを使用して指定: ターゲットシステムの選択	38
図 29	[Scan Devices...]の選択	39
図 30	I/Oデバイスの自動検出: 統合するデバイスの選択	39
図 31	TwinCAT 2 System Managerでのコンフィグレーションのマッピング	40
図 32	デバイスに接続されている個々のターミナルのスキャン	40
図 33	スタートアップ後のTwinCAT PLC Control	41
図 34	コンパイル処理後の変数付きのサンプルプログラム(変数統合なし)	41
図 35	TwinCAT PLC Controlプロジェクトの追加	42
図 36	System ManagerのPLCコンフィグレーションに追加されたPLCプロジェクト	42
図 37	PLC変数とプロセスオブジェクト間のリンクの作成	43
図 38	BOOL型のPDOの選択	43
図 39	複数のPDOの同時選択: [Continuous]および[All types]を有効化	44
図 40	「MAIN.bEL1004_Ch4」をサンプルとして使用した、「Goto Link」変数の使用例	44
図 41	ターゲットシステムの選択(リモート)	45
図 42	PLC制御にログイン、プログラムを開始可能	46
図 43	デフォルトのTwinCAT 3ユーザインターフェイス	47
図 44	TwinCATプロジェクトの新規作成	47
図 45	プロジェクトフォルダエクスプローラ内の新規TwinCAT3プロジェクト	48
図 46	選択ダイアログ: ターゲットシステムの選択	48

図 47	アクセスするPLCをTwinCAT System Managerを使用して指定: ターゲットシステムの選択...	49
図 48	[Scan]の選択.....	49
図 49	I/Oデバイスの自動検出: 統合するデバイスの選択.....	50
図 50	TwinCAT 3環境のVSシェルでのコンフィギュレーションのマッピング.....	50
図 51	デバイスに接続されている個々のターミナルのスキャン.....	51
図 52	[PLC]内でのプログラミング環境の追加.....	52
図 53	PLCプログラミング環境の名前およびディレクトリの指定.....	52
図 54	標準PLCプロジェクトの初期「Main」プログラム.....	53
図 55	コンパイル処理後の変数付きのサンプルプログラム(変数統合なし).....	54
図 56	プログラムのコンパイルの開始.....	54
図 57	PLC変数とプロセスオブジェクト間のリンクの作成.....	55
図 58	BOOL型のPDOの選択.....	55
図 59	複数のPDOの同時選択: [Continuous]および[All types]を有効化.....	56
図 60	「MAIN.bEL1004_Ch4」をサンプルとして使用した、「Goto Link」変数の使用例.....	56
図 61	TwinCAT開発環境(VSシェル): プログラム起動後にログイン.....	57
図 62	System Managerの[Options] (TwinCAT 2).....	58
図 63	VSシェルでの起動(TwinCAT 3).....	59
図 64	ネットワークインターフェースの概要.....	59
図 65	EtherCATデバイスのプロパティ(TwinCAT 2): タブ[Adapter]の[Compatible Devices]をクリック.....	59
図 66	ネットワークインターフェースのWindowsプロパティ.....	60
図 67	イーサネットポートに対する正しいドライバ設定の例.....	60
図 68	イーサネットポートに対する誤ったドライバ設定.....	61
図 69	イーサネットポートのTCP/IP設定.....	62
図 70	識別子の構造.....	63
図 71	オンラインディスクリプション情報ウィンドウ(TwinCAT 2).....	64
図 72	オンラインディスクリプション情報ウィンドウ(TwinCAT 3).....	64
図 73	System Managerによって作成されたOnlineDescription.xmlファイル.....	64
図 74	EL2521のオンラインで記録されたESI表示の例.....	65
図 75	ESIファイルに問題があった場合に表示される情報ウィンドウ(左: TwinCAT 2、右: TwinCAT 3).....	65
図 76	EtherCATデバイスの追加(左: TwinCAT 2、右: TwinCAT 3).....	66
図 77	EtherCAT接続の選択(TwinCAT 2.11、TwinCAT 3).....	66
図 78	イーサネットポートの選択.....	66
図 79	EtherCATデバイスのプロパティ(TwinCAT 2).....	67
図 80	EtherCATデバイスの追加(左: TwinCAT 2、右: TwinCAT 3).....	67
図 81	EtherCATデバイスの選択ダイアログ.....	68
図 82	デバイスのリビジョンの表示.....	68
図 83	以前のリビジョンの表示.....	69
図 84	ターミナルの名前/リビジョン.....	69
図 85	TwinCATツリー内のEtherCATターミナル(左: TwinCAT 2、右: TwinCAT 3).....	70
図 86	ローカルシステムとターゲットシステムの違い(左: TwinCAT 2、右: TwinCAT 3).....	71
図 87	デバイスのスキャン(左: TwinCAT 2、右: TwinCAT 3).....	71
図 88	自動デバイススキャンに関する注意(左: TwinCAT 2、右: TwinCAT 3).....	71
図 89	検出されたイーサネットデバイス.....	72
図 90	デフォルト状態の例.....	72
図 91	リビジョン1018のEtherCATターミナルのインストール.....	72
図 92	リビジョン-1019のEtherCATターミナルの検出.....	73
図 93	EtherCATデバイスの自動作成後のスキャン開始確認ダイアログ(左: TwinCAT 2、右: TwinCAT 3).....	73
図 94	指定したEtherCATデバイスに対するデバイススキャンの手動操作(左: TwinCAT 2、右: TwinCAT 3).....	73

図 95	TwinCAT 2によるスキヤンの進捗の例	74
図 96	Config/FreeRun確認ダイアログ(左: TwinCAT 2、右: TwinCAT 3)	74
図 97	右下のステータスバーでの「FreeRun」と「Config Mode」の交互表示	74
図 98	ボタンを使用してTwinCATをこの状態に切り替えることも可能(左: TwinCAT 2、右: TwinCAT 3)	74
図 99	オンライン表示の例	74
図 100	識別の失敗	75
図 101	同一のコンフィグレーション(左: TwinCAT 2、右: TwinCAT 3)	75
図 102	修正ダイアログ	76
図 103	ターミナルの名前/リビジョン	77
図 104	修正ダイアログでの変更	77
図 105	ダイアログ[Change to Compatible Type...] (左: TwinCAT 2、右: TwinCAT 3)	77
図 106	TwinCAT 2のダイアログ[Dialog Change to Alternative Type]	78
図 107	ターミナルEL3751としての分岐エレメント	78
図 108	[General]タブ	78
図 109	[EtherCAT]タブ	79
図 110	[Process Data]タブ	80
図 111	プロセスデータの設定	81
図 112	[Startup]タブ	81
図 113	[CoE - Online]タブ	83
図 114	[Advanced settings]ダイアログ	84
図 115	[Online]タブ	84
図 116	[DC]タブ(ディストリビュートクロック)	85
図 117	EtherCATスレーブの診断情報の選択	87
図 118	PLCによる基本的なEtherCATスレーブ診断	88
図 119	EL3102、CoEディレクトリ	90
図 120	EL3204のコミッショニング支援の例	91
図 121	System Managerのデフォルト動作	92
図 122	スレーブでのデフォルトのターゲット状態	92
図 123	PLCファンクションブロック	93
図 124	Eバス電流の超過	94
図 125	超過したEバス電流の警告メッセージ	94
図 126	ブリッジの基本機能	105
図 127	「スキヤン」後: ターミナルセグメント内のEL6695(左)、EtherCAT経由(RJ-45 X1経由)のボックス(右)	106
図 128	EL6695互換モード(デフォルトPDO)	107
図 129	デバイスからのPDOデータのロードによりエントリ「Active TX PDOs Map」を作成	108
図 130	PDOの構造	108
図 131	データスループット: 標準の場合と最適化を行った場合	109
図 132	EL6695のデータ転送シーケンス	110
図 133	「IO inputs」のコンテキストメニュー: 新しい変数の追加	111
図 134	ダイアログで名前がVar89から始まる新しい変数(ここでは4つのBYTE)を追加	111
図 135	EL6695のセカンダリに新しく追加された変数	111
図 136	プロセスデータの作成	112
図 137	EL6695のプライマリに適用した変数の例	113
図 138	[EL6695]タブ内の[Process Data]の[Create configuration]	114
図 139	[Load PDO info from device]でプライマリの変数をセカンダリにミラーリング転送	115
図 140	0x1C12による1バイト変数の0x7000「PD Outputs」への基本的なマッピング	116
図 141	選択的マッピングでのプロセスデータ	117
図 142	出力変数の新規作成(この例ではEL6695のプライマリ)	117
図 143	0x1614オブジェクトのセットアップ	118

図 144	グローバルデータ型への変換	119
図 145	RxPDO用のオブジェクト0x1C12の作成	120
図 146	プライマリの新規PDO Mappingおよび割り当てオブジェクト	120
図 147	新規作成されたプロセスデータ:オブジェクト0x1608	121
図 148	選択的に作成された変数および対応する参照の表示[CoE Online]	122
図 149	選択的PDO変数の追加	123
図 150	上書き用のスタートアップエントリ0x1A08(参照インデックスを修正01 → 02、02 → 03)	123
図 151	開始/停止バイトの使用	124
図 152	AoE ADSを使用した操作でのCoE書き込みアクセスを示す図	126
図 153	NetIDの採用	127
図 154	EoEコンフィグレーション	128
図 155	オブジェクト0xF800の正しい値(0x0000)の検証: EoE有効	129
図 156	EoE IPアドレスのチェック	129
図 157	コマンドラインを使用したマスタ1からマスタ2への「ping」の実行	130
図 158	ADSフレームルーティングの設定	131
図 159	AoE (ADS over EtherCAT)用のNetIDの設定	131
図 160	EL6695によるFoEデータ交換	134
図 161	CX2100-0014電源ユニットおよびEL6695を含むCX2040での構成	134
図 162	セカンダリへの0x1A05 (Data Bytes Pending)の追加	135
図 163	デバイスコンフィグレーション0xF800:01内に値0x4000を入力(コメント = Config 1: Disable EoE)	136
図 164	デバイスコンフィグレーション0xF800に両方の値を入力	137
図 165	プログラム例の状態図: 「OPEN」、「ACCESS」、および「CLOSE」によるランダムデータの書き込みおよび読み取り	138
図 166	*. tzipアーカイブを開く	140
図 167	サンプルのEtherCATコンフィグレーションで既存のHWコンフィグレーションを検索	141
図 168	EL6695でのFoEスループットテスト: データパケットサイズおよびデータ量ごとのタスクサイクル数	143
図 169	EL6695のFoEデータスループットの例について	144
図 170	ディストリビュートクロック: 同期なし	145
図 171	ディストリビュートクロック: 同期あり	145
図 172	タイムスタンプPDO 0x1A02または0x1A03の表示	145
図 173	ディストリビュートクロック: 外部リファレンスクロックの選択(この例ではプライマリ)	146
図 174	ディストリビュートクロック: 「SyncMaster」としてのEL3104の選択(この例ではセカンダリ)	146
図 175	ディストリビュートクロック: エラーメッセージ	147
図 176	24ビットのCoEオブジェクト0x1608	148
図 177	名前「EL3204-0000」およびリビジョン「-0016」から成るデバイス識別子	151
図 178	EtherCATデバイスを右クリックして下層のフィールドデバイスをスキャン	151
図 179	設定が同一	152
図 180	変更ダイアログ	152
図 181	EEPROM Update	153
図 182	新規ESIの選択	153
図 183	EL3204のファームウェアバージョンの表示	154
図 184	ファームウェア更新	155
図 185	FPGAファームウェアバージョン定義	157
図 186	コンテキストメニュー[Properties]	157
図 187	[Advanced settings]ダイアログ	158
図 188	複数選択およびファームウェア更新	160
図 189	[Online]タブによるファームウェア更新	161
図 190	BOOT状態の設定	161
図 191	LEDで動作状態を表示	162

図 192	「Restore default parameters」 PDIの選択.....	163
図 193	[Set Value]ダイアログでの復元値の入力.....	163