

EC-1 シリーズ

R01AN3853JJ0110

Rev.1.10

通信ボード EtherCAT®マニュアル

2018.09.04

要旨

産業イーサネット通信用 LSI EC-1 シリーズの通信評価ユニットを用いた EtherCAT 通信の使用方法について記載しております。

目次

1. 概要	3
1.1 構成	3
1.2 開発環境	4
1.2.1 ツールチェーン	4
1.2.2 EtherCAT Slave Stack Code Tool	4
1.2.3 TwinCAT	4
1.3 メモリ配置	5
1.4 プログラム配置例	6
2. ファイル構成	7
2.1 ディレクトリ構成	7
2.2 ./Include : インクルードファイル	7
2.3 ./Library : ライブラリ	7
2.4 ./Source : ソース	7
2.4.1 ./Source/Driver : ドライバ関連	7
2.4.2 ./Source/Project/EtherCAT_ComB : EtherCAT サンプルアプリケーション	8
2.4.3 ./Source/Project/EtherCAT_ComB_CiA402 : EtherCAT CiA402 サンプルアプリケーション	9
2.4.4 ./Source/Templates : スタートアップファイル等	10
3. TwinCAT マスタ通信	11
3.1 SSC サンプルソフト生成	11
3.2 サンプルソフトのプログラムダウンロード	16
3.2.1 EWARM 環境におけるプロジェクト起動およびビルド	16
3.2.2 e2studio 環境におけるプロジェクト起動およびビルド	18
3.3 TwinCAT 接続	21
4. サンプルアプリケーション操作方法	26
4.1 EtherCAT サンプルアプリケーション	26
4.2 EtherCAT CiA402 サンプルアプリケーション	28
5. Appendix A EtherCAT ドライバ・インストール	31
6. Appendix B EEPROM プログラム	33
ホームページとサポート窓口	35

1. 概要

ソフトウェア開発を速やかに進められるよう、EC-1 では各機能の使用例を示したサンプル・ソフトウェア（以降サンプルソフト）を準備しています。

本書では、EtherCAT[®]サンプルソフト構成、および TwinCAT[®]マスタとの通信手法について記しています。

その他のペリフェラルドライバについては、「EC-1 シリーズ アプリケーションノート ペリフェラルドライバマニュアル」(R01AN3581JJxxx)を参照してください。

1.1 構成

サンプルソフトのレイヤ構成図を以下に示します。

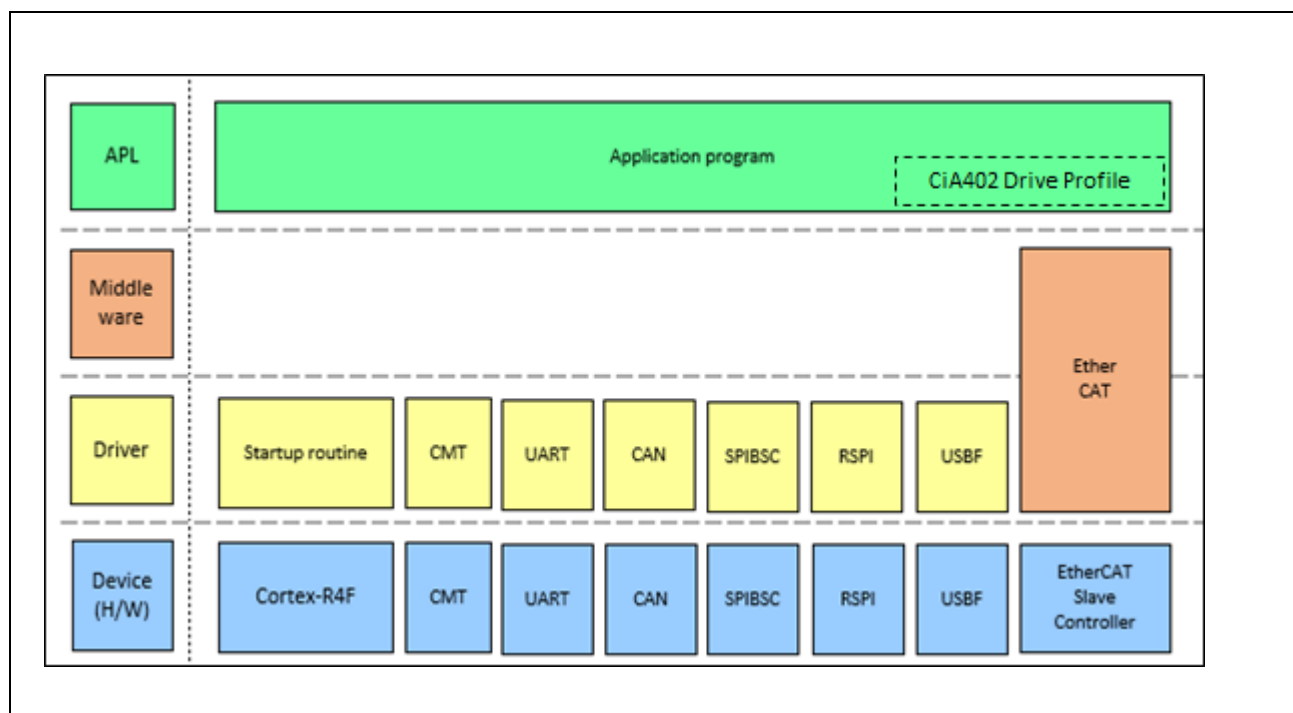


図 1-1 サンプルソフトのレイヤ構成図

1.2 開発環境

ソフトウェア開発ツールの説明を以下に示します。

1.2.1 ツールチェーン

本サンプルソフトウェアのツールチェーンは以下の通りです。

表 1-1 ソフトウェア開発ツール一覧（ツールチェーン）

ツール チェーン	IDE	コンパイラ	デバッガ	ICE
IAR	Embedded Workbench for ARM V7.70.1 以降（最新版をご使用下さい） (IAR Systems)			I-jet JTAGjet-Trace-CM (IAR Systems)
GCC	e2 studio V5.4.0 (Renesas Electronics)	KPIT GNUARM-NONE-EABI Toolchain v16.01 (KPIT Technologies)		J-Link (SEGGER)

1.2.2 EtherCAT Slave Stack Code Tool

EtherCAT サンプルソフトでは、EtherCAT Slave Stack Code Tool で生成した EtherCAT Slave Stack Code を使用します。EtherCAT Slave Stack Code Tool の入手には、EtherCAT Technology Group への加盟・Vendor ID の取得が必要です。

EtherCAT Slave Stack Code Tool については、EtherCAT Technology Group へお問い合わせ下さい。

<https://www.ethercat.org/>

1.2.3 TwinCAT

本アプリケーションノートでは、EEPROM への書き込み、及び EtherCAT サンプルソフトの動作確認に Beckhoff Automation 社の TwinCAT を使用します。

TwinCAT は Beckhoff Automation 社 Web サイトより入手可能です。

<http://www.beckhoff.com/>

1.3 メモリ配置

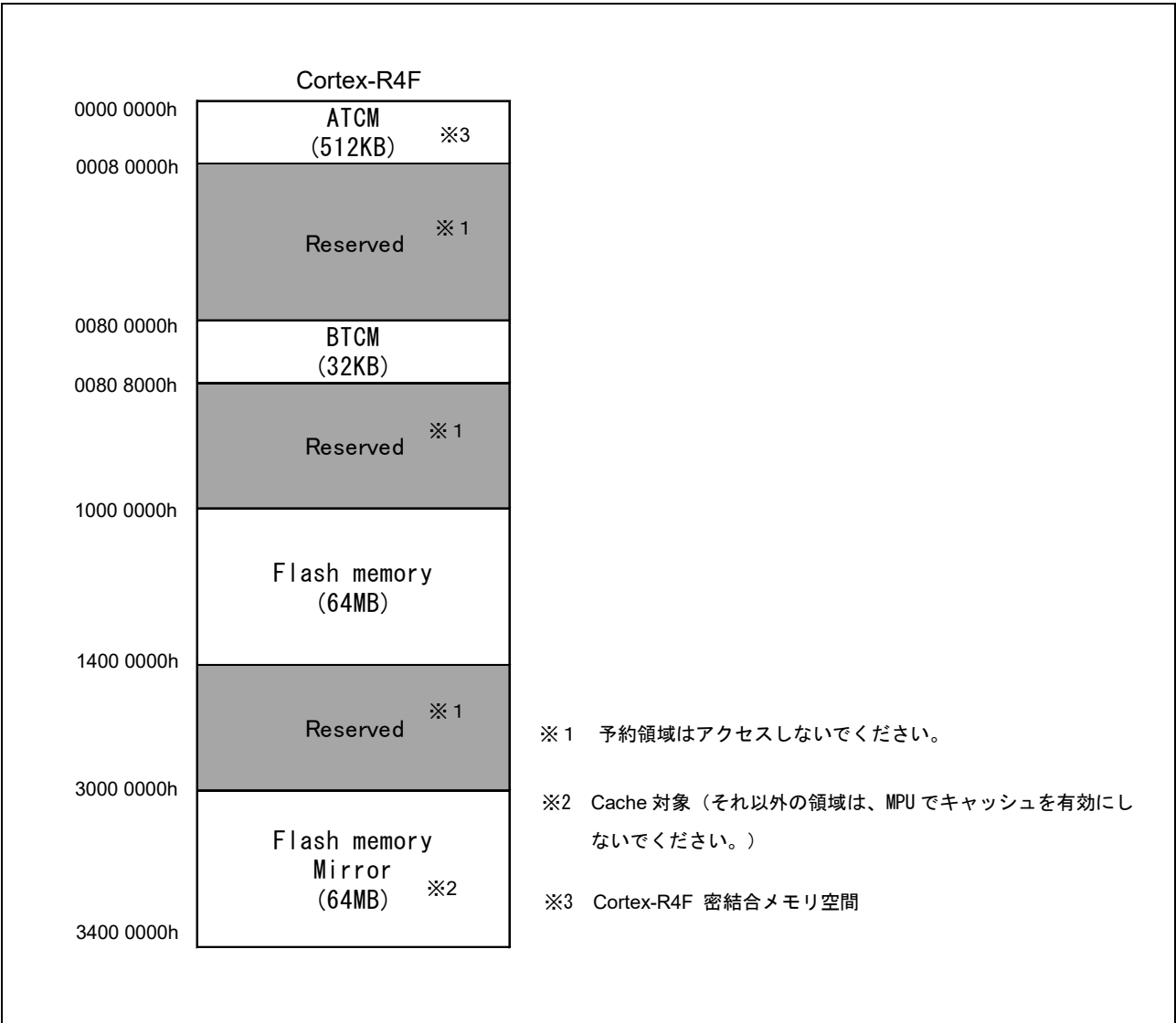


図 1-2 メモリマップ

1.4 プログラム配置例

シリアル・フラッシュ ROM ブート時のプログラム配置例を以下に示します。

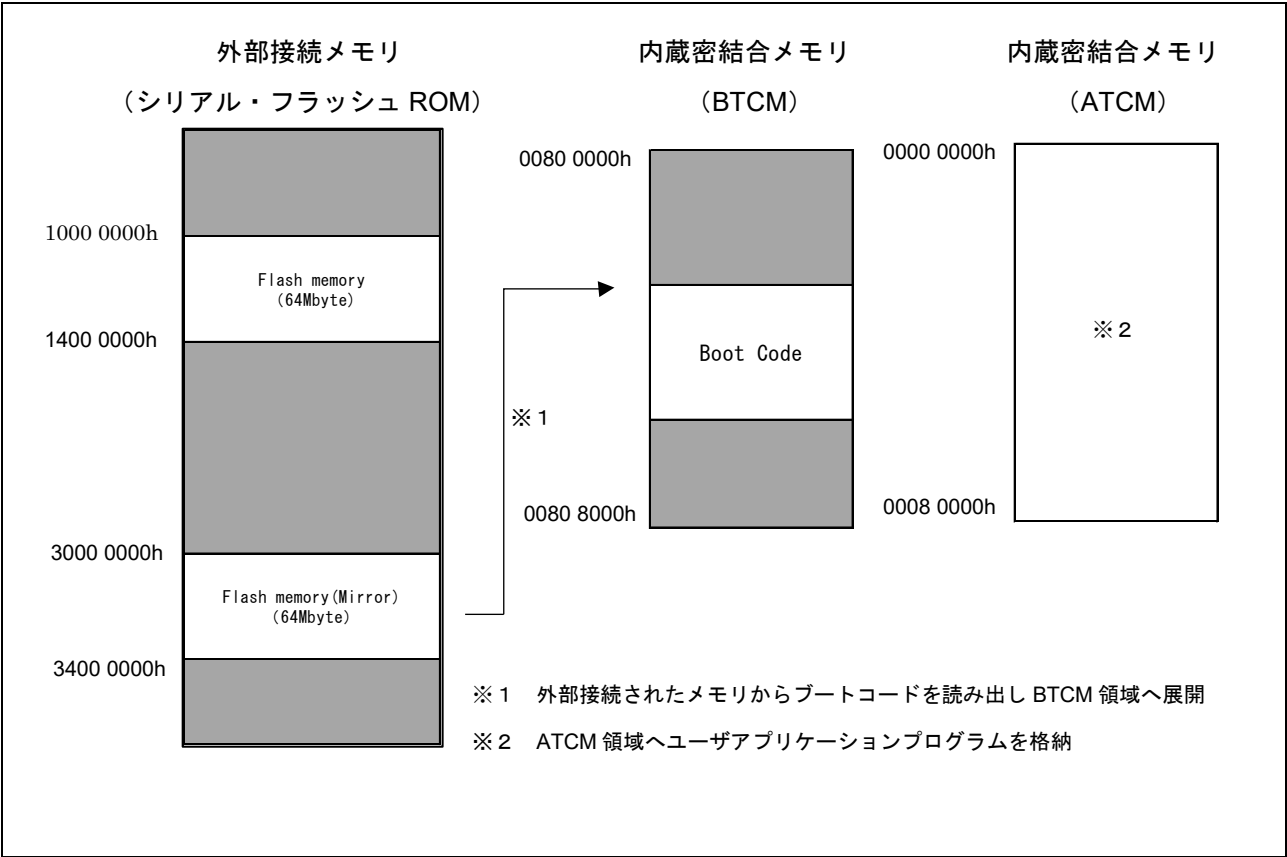


図 1-3 プログラム配置例

詳細は「EC-1 ユーザーズマニュアル ハードウェア編 3.4 動作モード」を参照して下さい。

2. ファイル構成

本章では、EtherCAT サンプルソフトのディレクトリ構成と EtherCAT 通信関連のファイル構成を示します。EtherCAT 通信以外の使用ファイルについては、「EC-1 シリーズ アプリケーションノート ペリフェラルドライバマニュアル」(R01AN3581JJxxxx) を参照してください。

2.1 ディレクトリ構成

表 2-1 サンプルソフトのディレクトリ構成

ディレクトリ	内容
./	サンプルソフト格納ディレクトリ
./Include	インクルードファイル格納ディレクトリ
./Library	ライブラリ格納ディレクトリ
./Source	ソース格納ディレクトリ

2.2 ./Include : インクルードファイル

以下に EtherCAT 通信に使用するインクルードファイルのファイル構成を示します。

表 2-2 インクルードファイルディレクトリのファイル構成

ディレクトリ	ファイル	内容
eth/	r_ether.h	ETHERドライバのプロトタイプ宣言

2.3 ./Library : ライブラリ

このディレクトリに EtherCAT 通信関連ファイルはありません。

2.4 ./Source : ソース

表 2-3 ソースディレクトリの構成

ディレクトリ	内容
Driver	ドライバ関連
Project	サンプルアプリケーション
Templates	スタートアップ・ファイル等

2.4.1 ./Source/Driver : ドライバ関連

以下に EtherCAT 通信に使用するドライバのソースファイルの構成を示します。

表 2-4 ドライバ関連ディレクトリのファイル構成

ディレクトリ	ファイル	内容
ether/	r_ether.c	ETHERドライバ

2.4.2 ./Source/Project/EtherCAT_ComB : EtherCAT サンプルアプリケーション

以下に EtherCAT サンプルアプリケーションのファイル構成を示します。

EtherCAT サンプルアプリケーションでは Slave Stack Code が必要です。

表 2-5 EtherCAT サンプルアプリケーションのファイル構成 ★

ディレクトリ	ファイル	内容
EtherCAT_ComB/	board_communication.c	ボード設定ファイル
	EC-1 ComB.esp	SSCプロジェクトファイル【SSC生成時に保存】
	EC-1 ComB.xml	EtherCAT Slave Information (ESI)ファイル 【SSC実行時に生成】
	main.c	メイン処理
	renesashw.c	ハードウェア処理
	renesashw.h	ハードウェア処理のプロトタイプ宣言
GCC/	EC-1_e2sws_serial_boot.bat	e2Studioプロジェクト起動バッチファイル
	serial_boot_sample.zip	e2Studioプロジェクトアーカイブファイル
IAR/	EC-1_Comb_ecat.ewd	IARプロジェクト関連ファイル
	EC-1_Comb_ecat.ewp	IARプロジェクト関連ファイル
	EC-1_Comb_ecat.eww	IAR EWARMプロジェクトファイル
SSC_config/	Renesas_EC-1_ComB_config.xml	SSC Tool コンフィギュレーションファイル
Src/ 【SSC実行後に生成】	applInterface.h	アプリケーション インタフェースのプロトタイプ宣言
	coeappl.c	CoEアプリケーション
	coeappl.h	CoEアプリケーションのプロトタイプ宣言
	ecat_def.h	EtherCAT 関連のプロトタイプ宣言
	ecatappl.c	アプリケーション
	ecatappl.h	アプリケーションのプロトタイプ宣言
	ecatcoe.c	CoE mailbox処理
	ecatcoe.h	CoE mailboxのプロトタイプ宣言
	ecatstv.c	EtherCAT State Machine処理
	ecatstv.h	EtherCAT State Machineのプロトタイプ宣言
	esc.h	EtherCAT Slave Controller (ESC) のプロトタイプ宣言
	mailbox.c	Mailbox処理
	mailbox.h	Mailboxのプロトタイプ宣言
	main.c	メイン処理
	objdef.c	Object Dictionary関連処理
	objdef.h	Object Dictionary関連処理のプロトタイプ宣言
	renesashw.c	ハードウェア処理
	renesashw.h	ハードウェア処理のプロトタイプ宣言
	sampleappl.c	サンプルアプリケーション
	sampleappl.h	サンプルアプリケーションのプロトタイプ宣言
	sdsoserv.c	SDO サービス処理
	sdsoserv.h	SDO サービス処理のプロトタイプ宣言

2.4.3 ./Source/Project/EtherCAT_ComB_CiA402 : EtherCAT CiA402 サンプルアプリケーション

以下に EtherCAT CiA402 サンプルアプリケーションのファイル構成を示します。

EtherCAT CiA402 サンプルアプリケーションでは Slave Stack Code が必要です。

表 2-6 EtherCAT CiA402 サンプルアプリケーションのファイル構成

ディレクトリ	ファイル	内容
EtherCAT_ComB_CiA402/	apply_patch.bat	Patch適用batファイル
	board_communication.c	ボード設定ファイル
	EC-1 CiA402.esp	SSCプロジェクトファイル【SSC生成時に保存】
	main.c	メイン処理
	renesashw.c	ハードウェア処理
	renesashw.h	ハードウェア処理のプロトタイプ宣言
	SSC_CiA402_yyyymmdd.patch	CiA402向け変更パッチファイル (yyyyymmdd : パッチファイル作成日)
ESI_File/	Renesas_EC-1_ComB_CiA402.xml	EtherCAT Slave Information (ESI)ファイル
GCC/	EC-1_e2sws_serial_boot.bat	e2Studioプロジェクト起動バッチファイル
	serial_boot_sample.zip	e2Studioプロジェクトアーカイブファイル
IAR/	EC-1_Comb_ecat_CiA402.ewd	IARプロジェクト関連ファイル
	EC-1_Comb_ecat_CiA402.ewp	IARプロジェクト関連ファイル
	EC-1_Comb_ecat_CiA402.eww	IAR EWARMプロジェクトファイル
SSC_config/	Renesas_EC-1_ComB_CiA402_config.xml	SSC Tool コンフィギュレーションファイル
Src/ 【SSC実行後に生成】	applInterface.h	アプリケーション インタフェースのプロトタイプ宣言
	cia402appl.c	サンプルアプリケーション
	cia402appl.h	サンプルアプリケーションのプロトタイプ宣言
	coeappl.c	CoEアプリケーション
	coeappl.h	CoEアプリケーションのプロトタイプ宣言
	ecat_def.h	EtherCAT 関連のプロトタイプ宣言
	ecatappl.c	アプリケーション
	ecatappl.h	アプリケーションのプロトタイプ宣言
	ecatcoe.c	CoE mailbox処理
	ecatcoe.h	CoE mailboxのプロトタイプ宣言
	ecatslv.c	EtherCAT State Machine処理
	ecatslv.h	EtherCAT State Machineのプロトタイプ宣言
	esc.h	EtherCAT Slave Controller (ESC) のプロトタイプ宣言
	mailbox.c	Mailbox処理
	mailbox.h	Mailboxのプロトタイプ宣言
	main.c	メイン処理
	objdef.c	Object Dictionary関連処理
	objdef.h	Object Dictionary関連処理のプロトタイプ宣言
	renesashw.c	ハードウェア処理
	renesashw.h	ハードウェア処理のプロトタイプ宣言
	sdoserv.c	SDO サービス処理
	sdoserv.h	SDO サービス処理のプロトタイプ宣言

2.4.4 ./Source/Templates : スタートアップファイル等

以下にスタートアップ・ファイル等のソースファイルの構成を示します。

表 2-7 スタートアップ関連ディレクトリのファイル構成

ディレクトリ	ファイル	内容
Templates/	exit.c	終了シーケンス
	r_atcm_init.c	ATCMアクセスウェイト設定API
	r_cpg.c	CPG設定API
	r_ecm.c	ECM設定API
	r_icu_init.c	EC-1各機器設定初期化
	r_mpc.c	MPC設定API
	r_reset.c	EC-1リセットAPI及び低電力API
Templates/IAR/	loader_init.asm	EC-1用割り込みサービスルーチン
	vector.asm	ベクタテーブル設定
Templates/IAR/serial_boot	bus_init_serial_boot.c	バス設定初期化
	EC-1_init_serial_boot.icf	マッピングファイル
	EC1_init_boot.mac	初期化マクロファイル
	loader_init_sflash.c	EC-1ペリフェラル設定初期化
	loader_param_serial_boot.c	SPIブートモード用パラメータ設定
Templates/IAR/ram_debug	EC-1_init_ram_debug.icf	マッピングファイル
	EC1_init_ram_debug.mac	初期化マクロファイル
	loader_init_ram.c	EC-1ペリフェラル設定初期化
Templates/GCC	loader_init.asm	EC-1用割り込みサービスルーチン
	vector.asm	ベクタテーブル設定
Templates/GCC/ram_debug	loader_init_ram.c	EC-1ペリフェラル設定初期化
Templates/GCC/serial_boot	bus_init_serial_boot.c	バス設定初期化
	loader_init_sflash.c	EC-1ペリフェラル設定初期化
	loader_param_serial_boot.c	SPIブートモード用パラメータ設定

3. TwinCAT マスタ通信

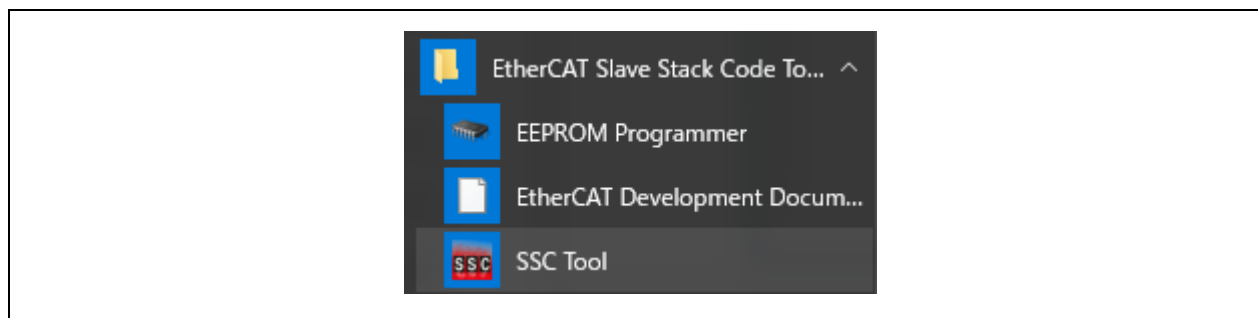
本章では、TwinCAT3 マスタとスレーブ EtherCAT 通信ボードとの通信手法を示します。

3.1 SSC サンプルソフト生成

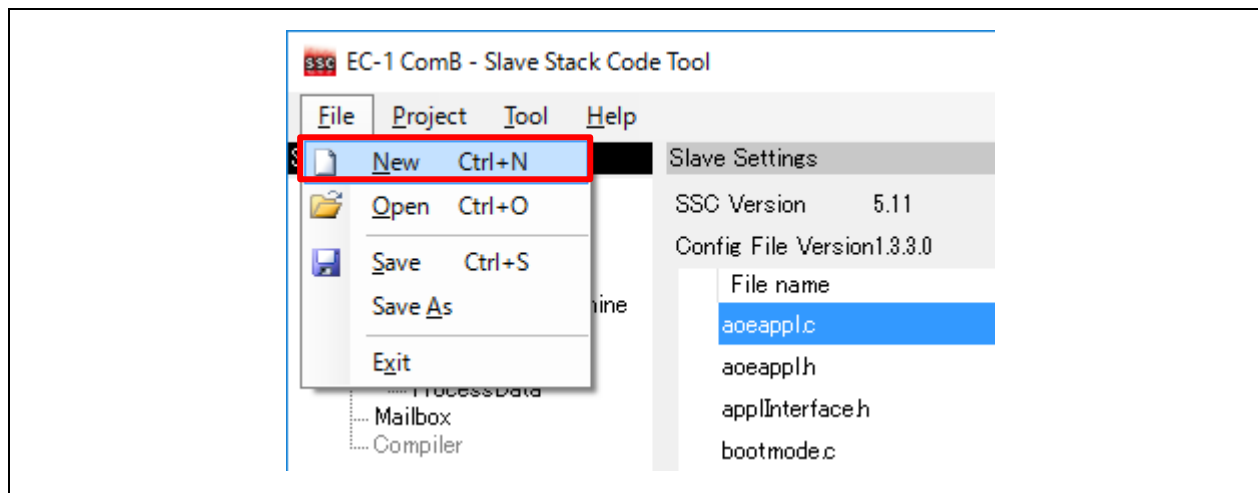
EtherCAT サンプルソフトの使用には、EtherCAT Slave Stack Code (SSC)が必要です。

以下の手順で SSC Tool を使用して生成します。

1. SSC Tool を Windows Start menu より起動



2. [File] > [New] 選択



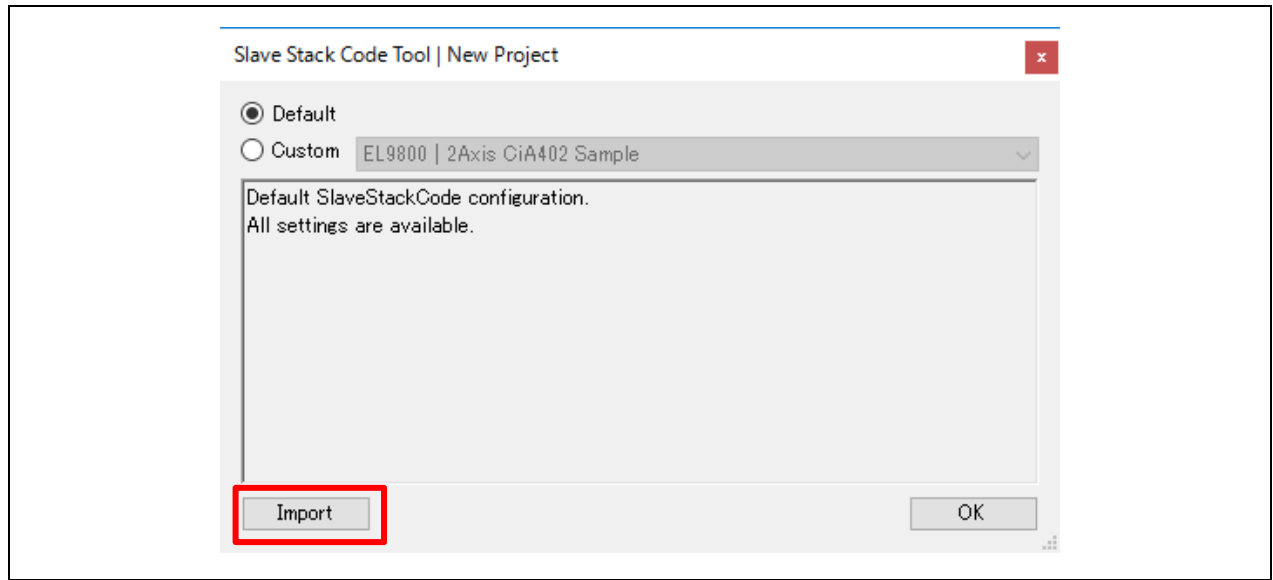
3. [Import] を押し、SSC Tool コンフィギュレーションファイルを選択

- ・ EtherCAT サンプルアプリケーション

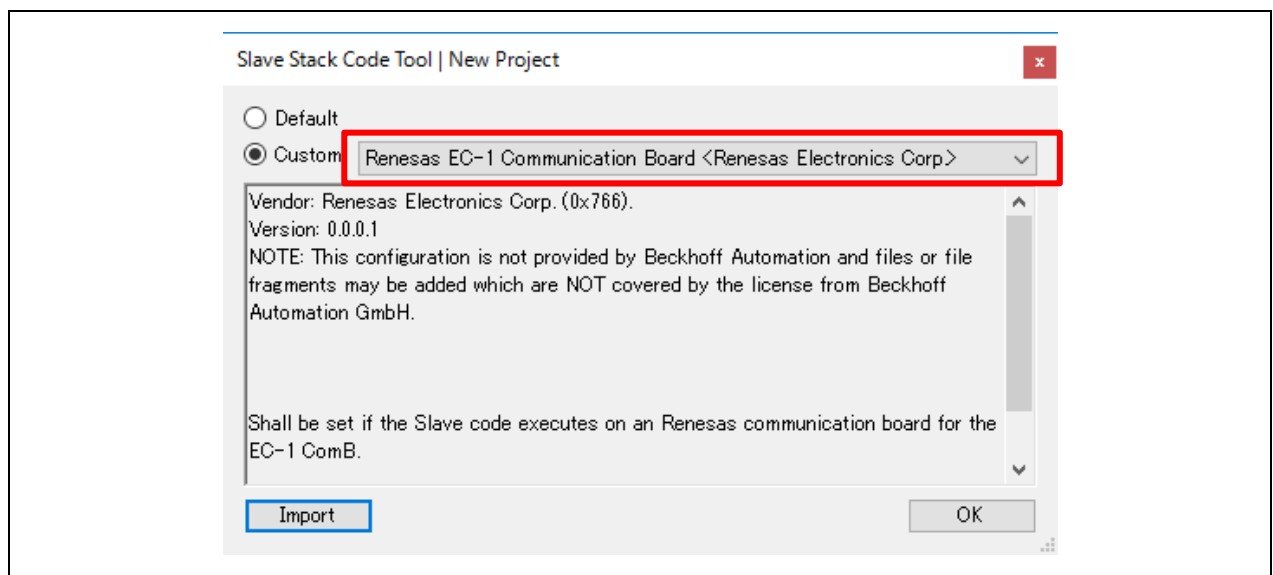
`./Project/EtherCAT_ComB/SSC_config/Renesas_EC-1_ComB_SSCconfig.xml`

- ・ EtherCAT CiA402 サンプルアプリケーション

`./Project/EtherCAT_ComB_CiA402/SSC_config/Renesas_EC-1_ComB_CiA402_config.xml`



4. コンフィギュレーションファイルが読み込まれると以下ようになります



一度読み込むと、以降は Custom に登録されドロップダウンより選択可能となります

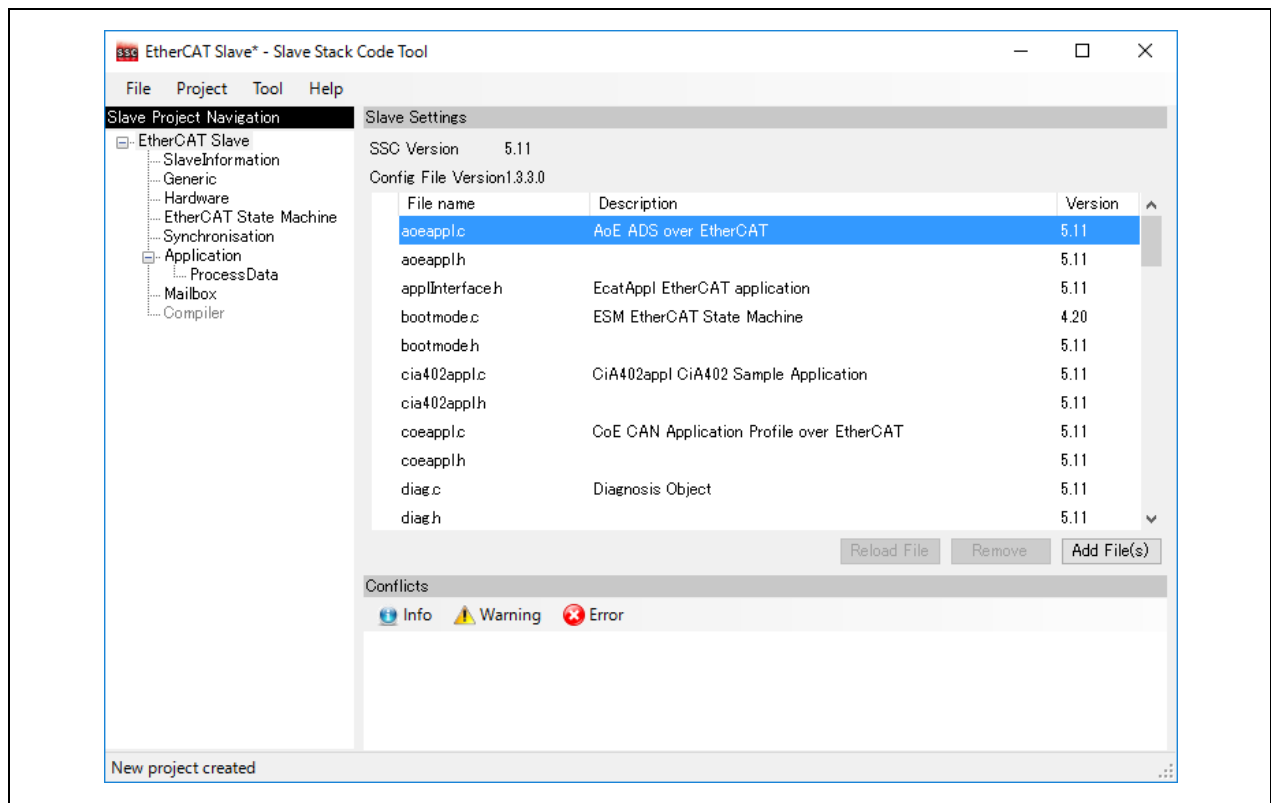
5. [OK]を押すと以下の画面が表示されます

- EtherCAT サンプルアプリケーション

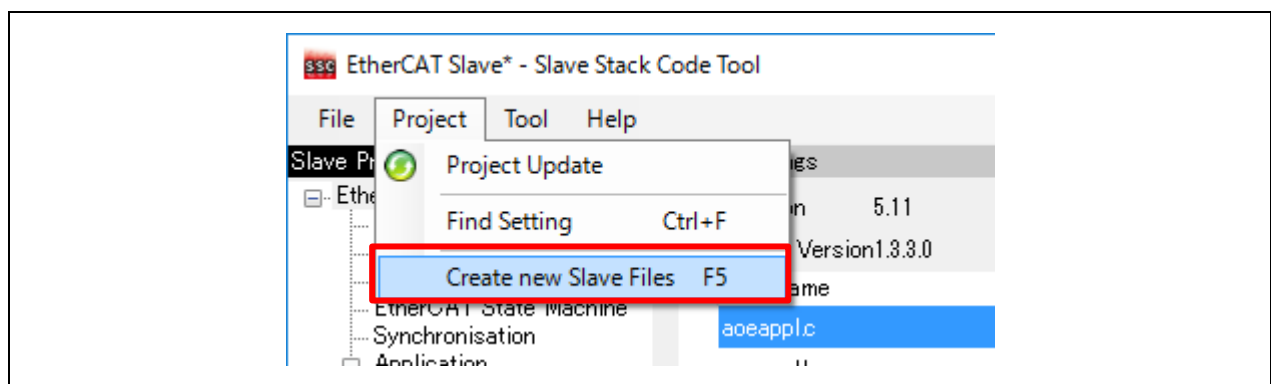
`./Project/EtherCAT_ComB/renesashw.c`

- EtherCAT CiA402 サンプルアプリケーション

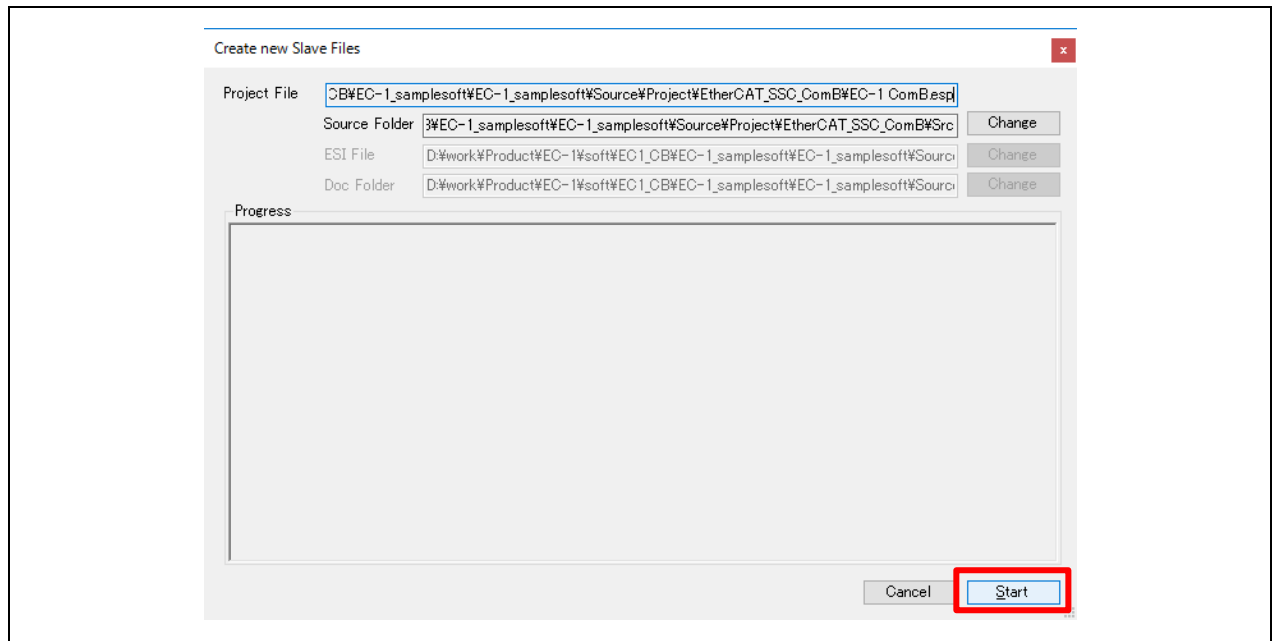
`./Project/EtherCAT_ComB_CiA402/renesashw.c`



6. [Project] > [Create new Slave Files] を選択



7. Start を押し、EtherCAT Slave Stack Code 生成を開始



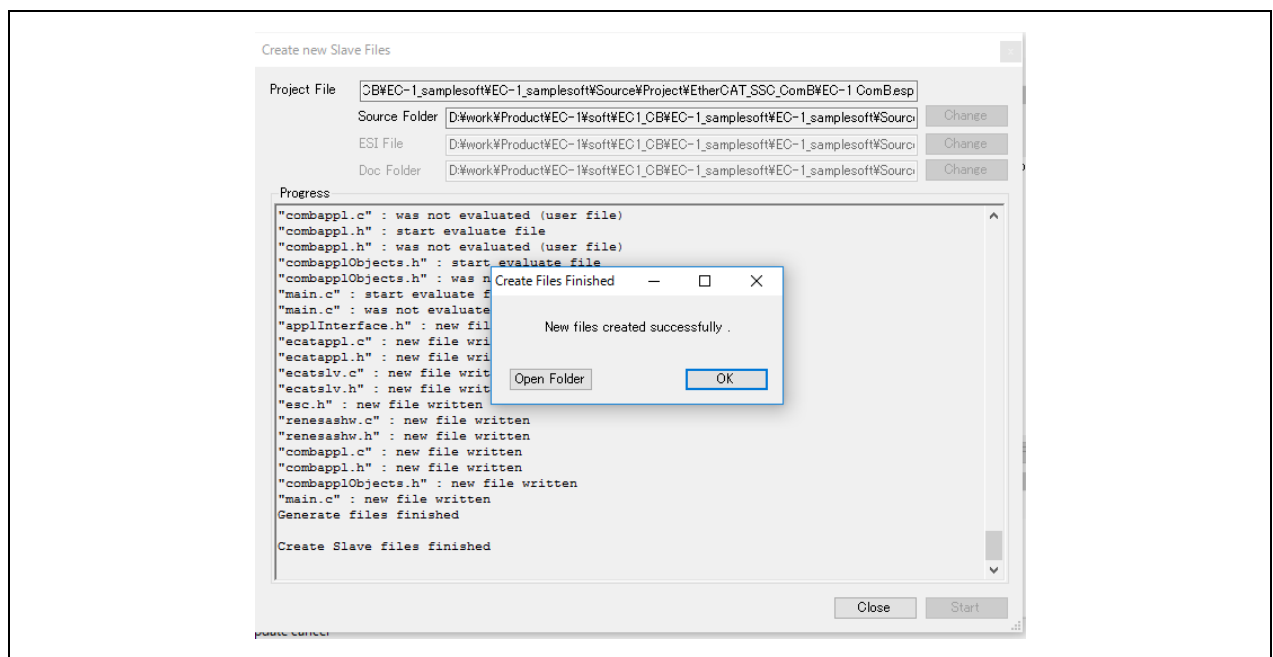
8. [New files created successfully] が表示されると生成完了し、Src ファイルが生成されます

- EtherCAT サンプルアプリケーション

`./Project/EtherCAT_ComB/Src`

- EtherCAT CiA402 サンプルアプリケーション

`./Project/EtherCAT_ComB_CiA402/Src`



9. bat ファイルを実行し、patch を適用 (EtherCAT CiA402 サンプルアプリケーションのみ)

bat ファイルが付属しているサンプルアプリケーションでは、bat ファイルを実行し、patch を適用します。

・ Patch Command をインストールしていない場合

下記の Link からファイルをダウンロードし、セットアップ、または patch.exe を path の通っている directory に格納して下さい。

<http://gnuwin32.sourceforge.net/packages/patch.htm>

・ Patch の適用

apply_patch.bat ファイルを右クリックして [管理者として実行] > [はい] を選択します。

これにより patch ファイルが実行され、SSC ソースファイルへサンプルアプリケーションの修正が適用されます。

・ EtherCAT サンプルアプリケーション

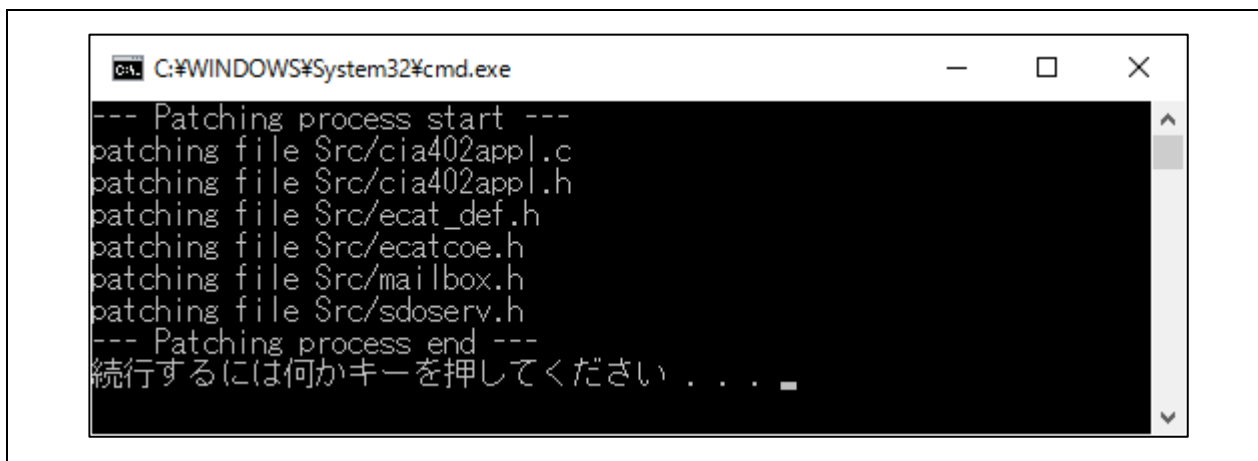
(不要)

・ EtherCAT CiA402 サンプルアプリケーション

./Project/EtherCAT_ComB_CiA402/apply_patch.bat (バッチファイル)

./Project/EtherCAT_ComB_CiA402/SSC_CiA402_yyyymmdd.patch (パッチファイル)

(yyymmdd: パッチファイル作成日)



3.2 サンプルソフトのプログラムダウンロード

生成された EtherCAT サンプルソフトを以下の手順に従い通信ボードへダウンロードします。

3.2.1 EWARМ 環境におけるプロジェクト起動およびビルド

1. ICE の JTAG コネクタを通信ボード CN2 へ接続
2. EWARМ 起動

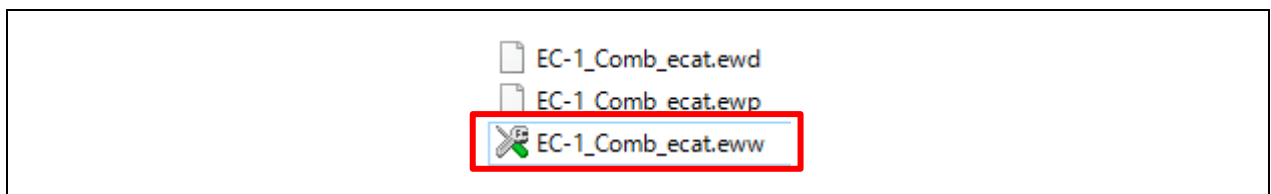
サンプルソフトの EWARМ プロジェクトファイルをダブルクリックし起動

- ・ EtherCAT サンプルアプリケーション

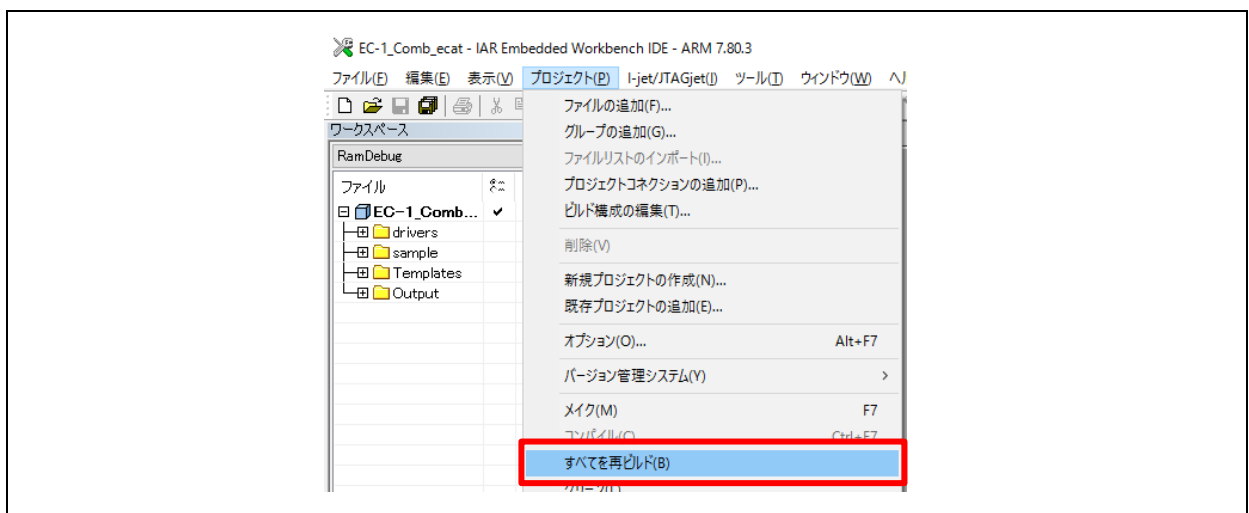
`./Source/Project/EtherCAT_ComB/IAR/EC-1_Comb_ecat.eww`

- ・ EtherCAT CiA402 サンプルアプリケーション

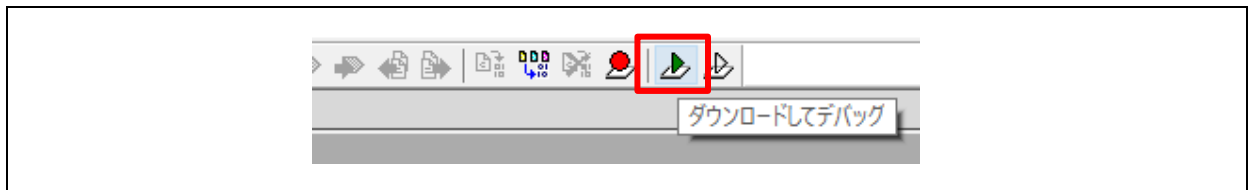
`./Source/Project/EtherCAT_ComB_CiA402/IAR/EC-1_Comb_ecat_CiA402.eww`



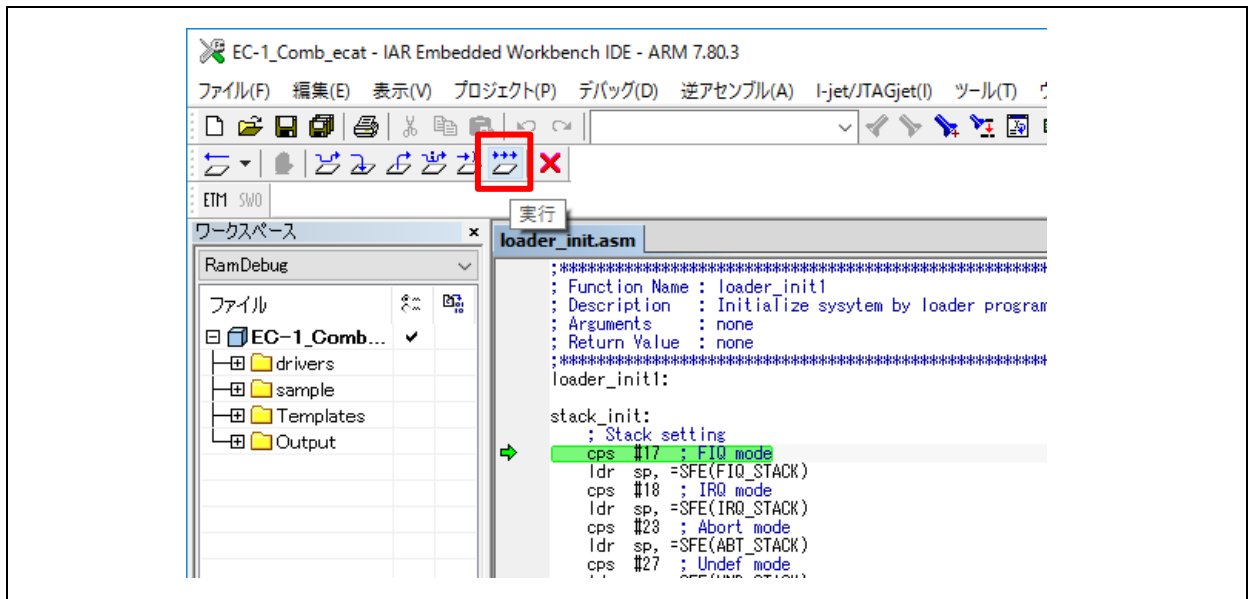
3. ビルド実行



4. [ダウンロードしてデバッグ]を実行



5. プログラム実行



3.2.2 e2studio 環境におけるプロジェクト起動およびビルド

1. ICE の JTAG コネクタを通信ボード CN2 へ接続
2. e2 studio 起動

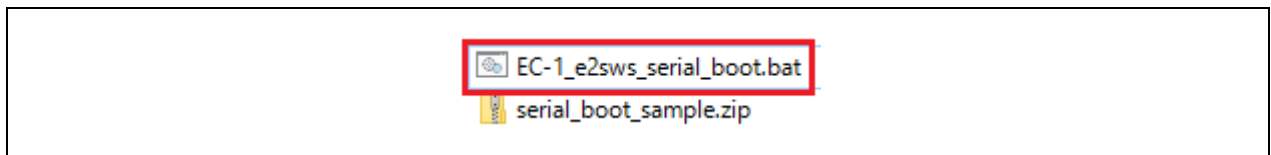
サンプルソフトのバッチファイルをダブルクリックし起動

- ・ EtherCAT サンプルアプリケーション

`./Source/Project/EtherCAT_ComB/GCC/EC-1_e2sws_serial_boot.bat`

- ・ EtherCAT CiA402 サンプルアプリケーション

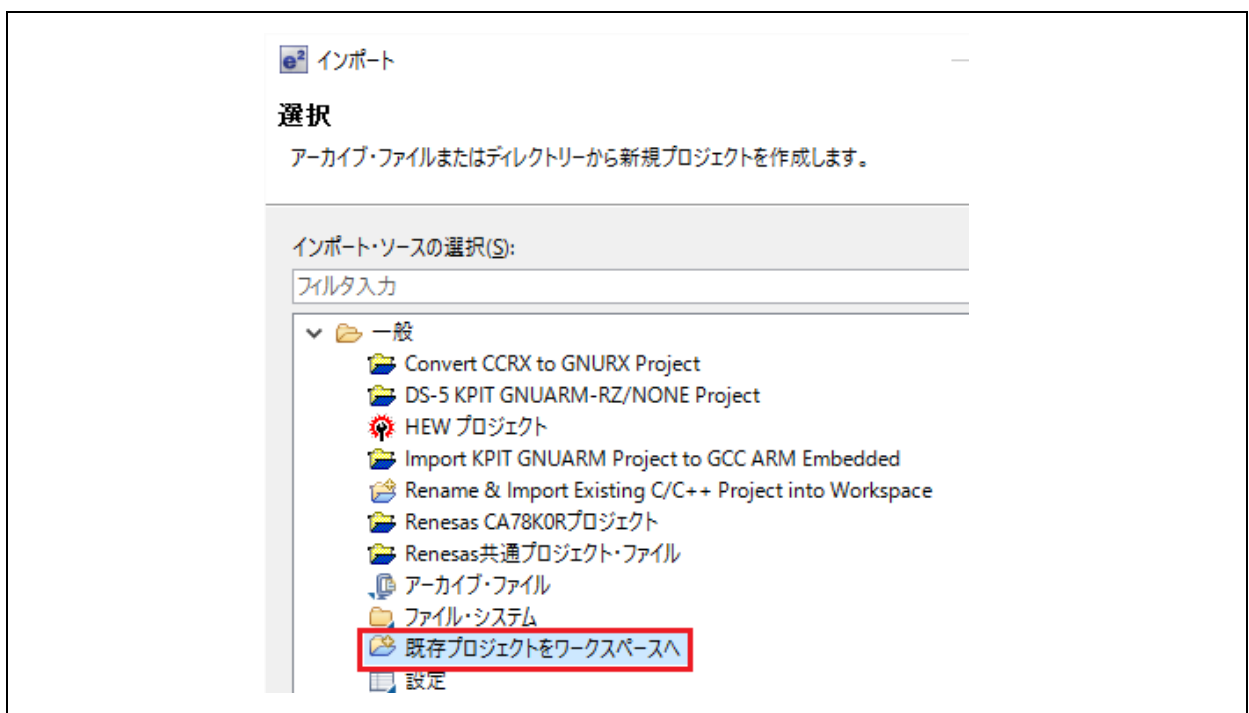
`./Source/Project/EtherCAT_ComB_CiA402/GCC/EC-1_e2sws_serial_boot.bat`



3. プロジェクトのインポート (初回のみ)

バッチファイルと同じフォルダに格納されているアーカイブ・ファイルをインポート

ファイル > インポート > 一般 > 既存プロジェクトをワークスペースへ > 次へ

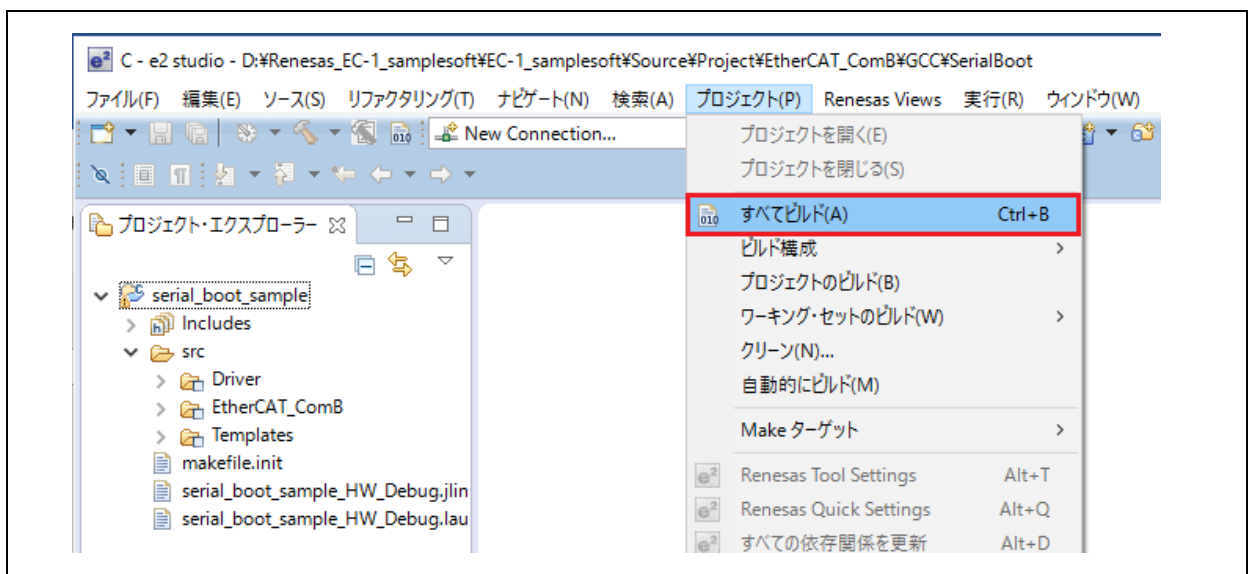


アーカイブ・ファイルの選択 > 参照 > (serial_boot_sample.zip を選択) > 終了



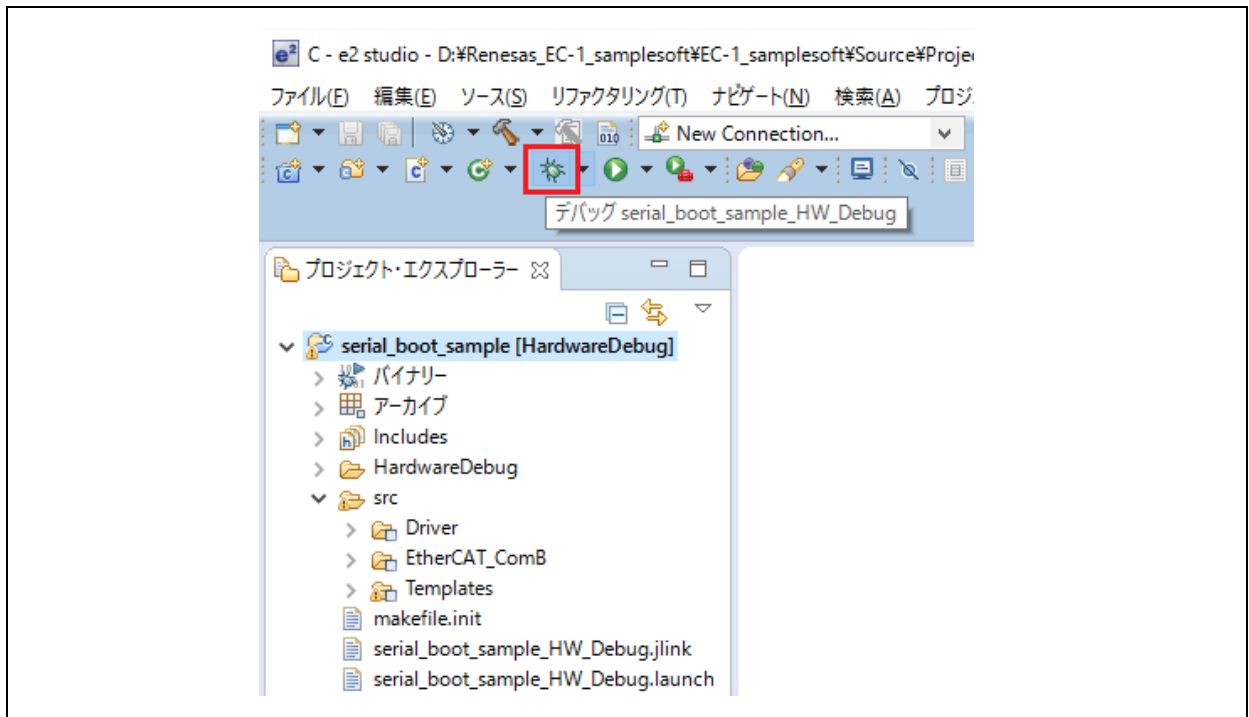
4. ビルド実行

プロジェクト > すべてのビルド



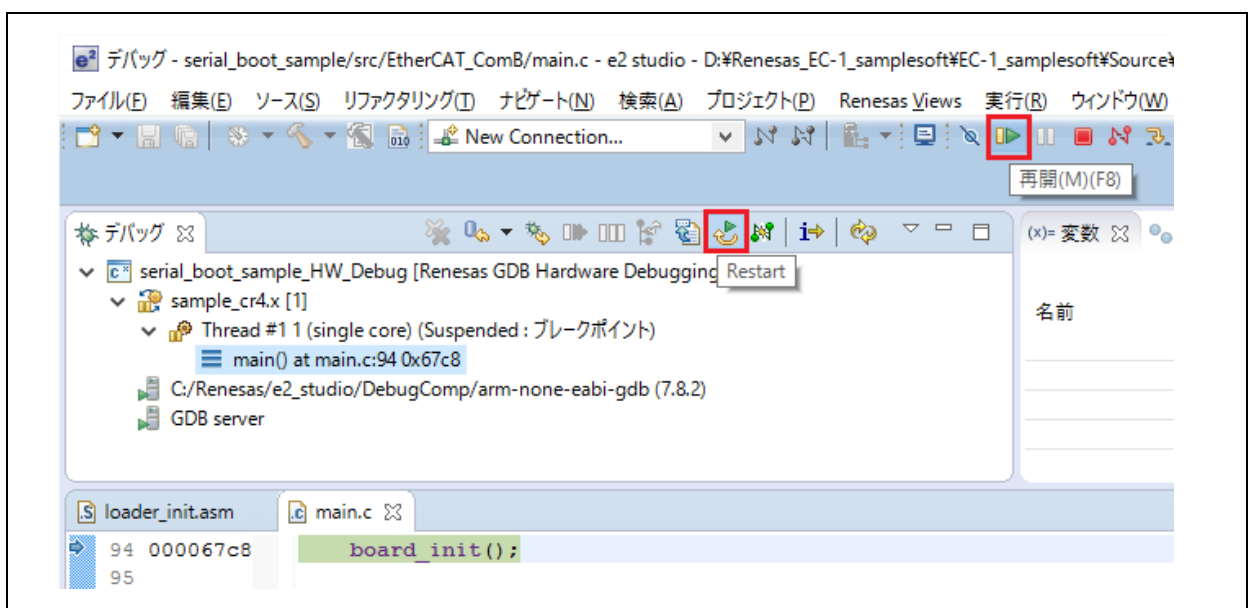
5. ダウンロード

プロジェクト・エクスプローラーの serial_boot_sample をクリックし、[HardwareDebug]が表示された後に「デバッグ」をクリック (次回から「デバッグ」のクリックのみ)



6. プログラム実行

Restart > 再開



3.3 TwinCAT 接続

TwinCAT3 と通信ボードの EtherCAT 接続手法を説明します。

1. ESI ファイル

通信ボード用 EtherCAT サンプルソフトの EtherCAT Slave Information (以下、ESI)ファイルを TwinCAT フォルダへ格納します。

【ESI ファイル】

- ・ EtherCAT サンプルアプリケーション(SSC 実行時に生成)

`./Source/Project/EtherCAT_ComB/EC-1 ComB.xml` ★

- ・ EtherCAT CiA402 サンプルアプリケーション

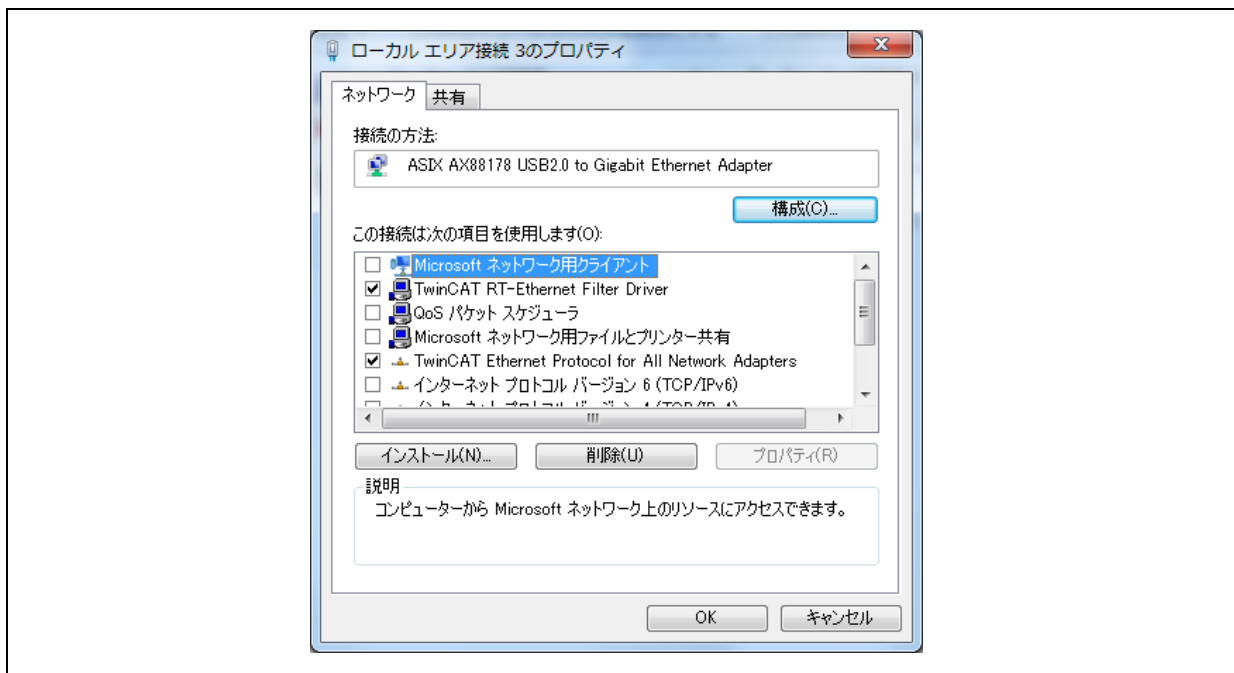
`./Source/Project/EtherCAT_ComB_CiA402/ESI_File/Reanasas_EC-1_ComB_CiA402.xml`

【格納先】

`C:\TwinCAT\3.1\Config\Io\EtherCAT`

2. ネットワークアダプタ設定

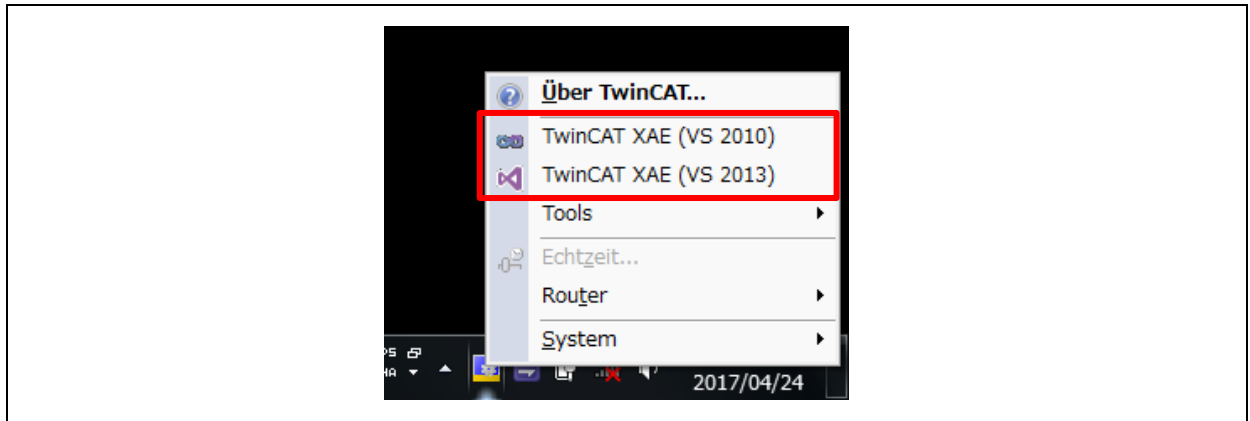
Windows のネットワーク接続 > TwinCAT として使用するネットワークアダプタのプロパティを表示し、[TwinCAT RT-Ethernet Filter Driver]、[TwinCAT Ethernet Protocol] のみを有効にします。



[TwinCAT RT-Ethernet Filter Driver]の表示が無い、もしくはドライバをインストールしていない場合は、Appendix A を参照してドライバをインストールして下さい。

3. TwinCAT 起動

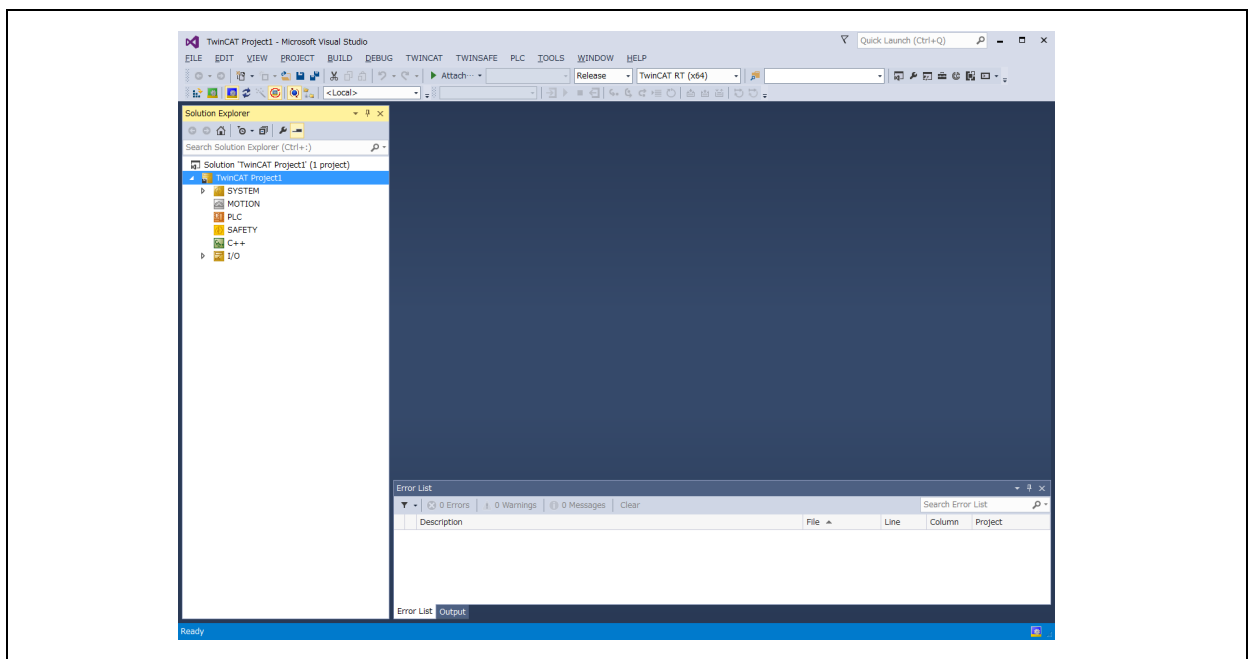
タスクから TwinCAT XAE (VS 20xx)をクリック



VS 2010, VS2013 どちらでも実行可能です。インストールされている Visual Studio のバージョンのみ表示されます。

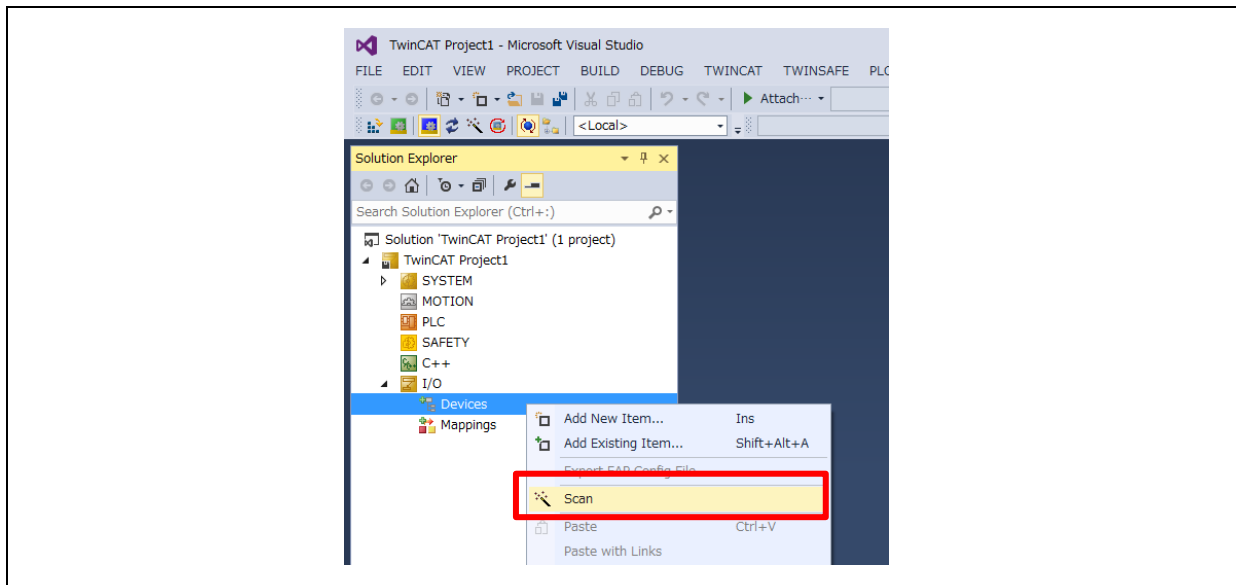
4. New TwinCAT Project 起動

[FILE] > [New] > [Project] を選択し、[OK]を押すと TwinCAT プロジェクトが表示されます



5. Scan 実行

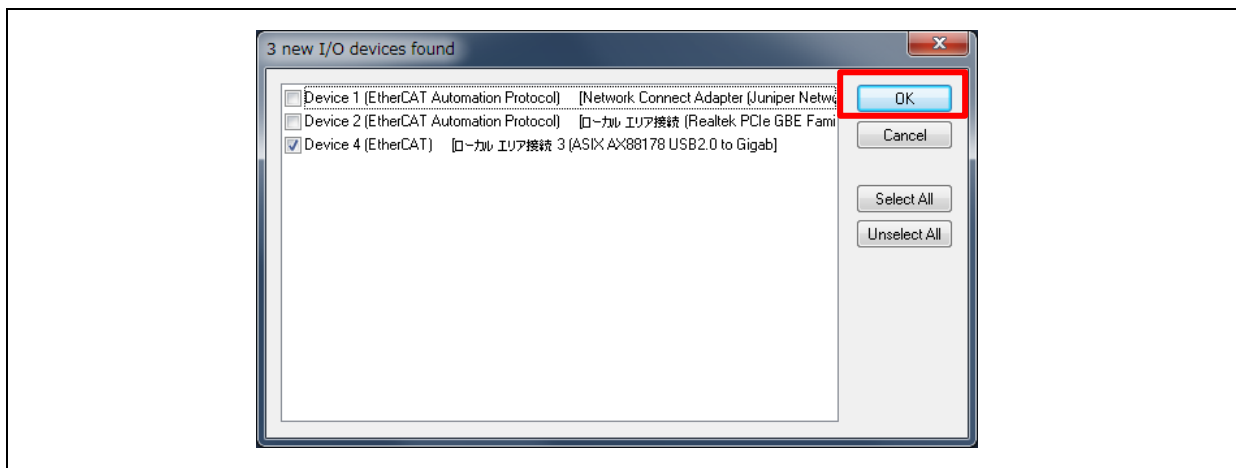
I/O > Devices を右クリックし、[Scan]を実行。



6. 使用するアダプタを選択

EtherCAT に対応したスレーブが接続されている場合は、自動で Device の横にチェックが入ります。

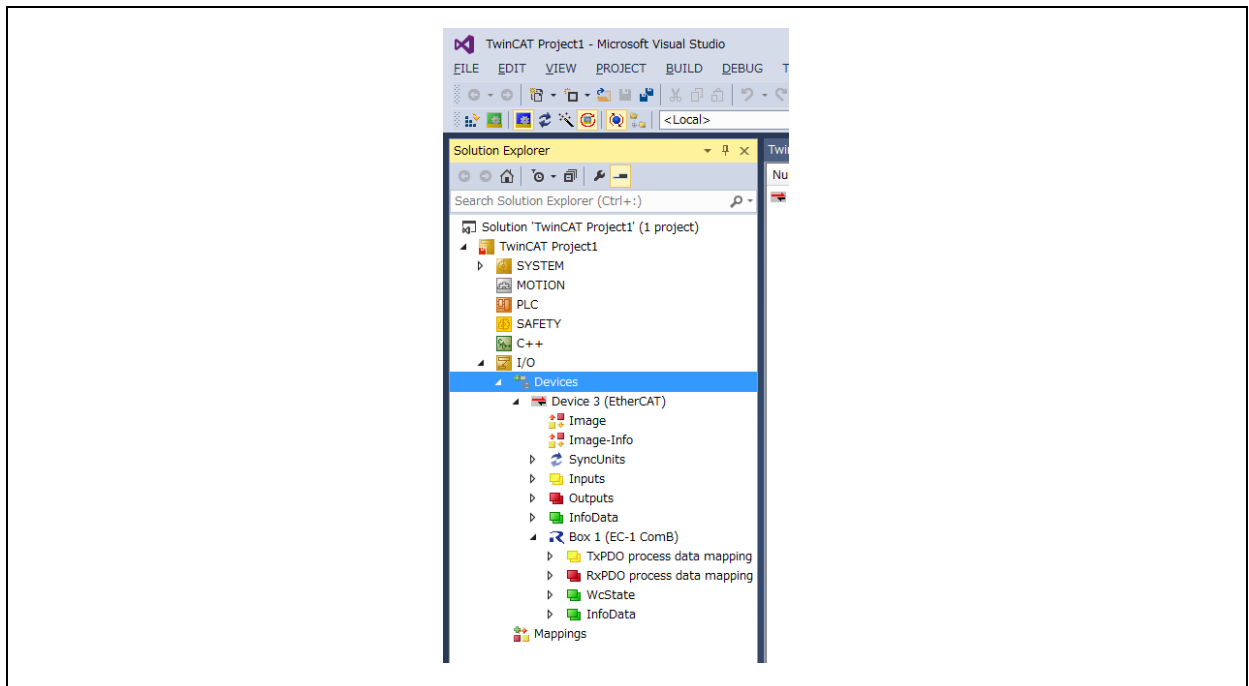
[OK] を押し、Scan for Box を実行。



[確認] 自動でチェックが付かない、または [EtherCAT]が表示されない場合は、以下の要因が考えられます。

- ・ 通信ボードとイーサケーブルが接続されていない
 - 通信ボードと TwinCAT がインストールされた PC をイーサで接続してください
- ・ ネットワークアダプタの設定が TwinCAT 設定になっていない
 - 本項の手順 2 を参照し、PC のネットワーク環境を変更してください
- ・ EtherCAT ドライバがインストールされていない
 - Appendix A を参照し、EtherCAT ドライバをインストールしてください

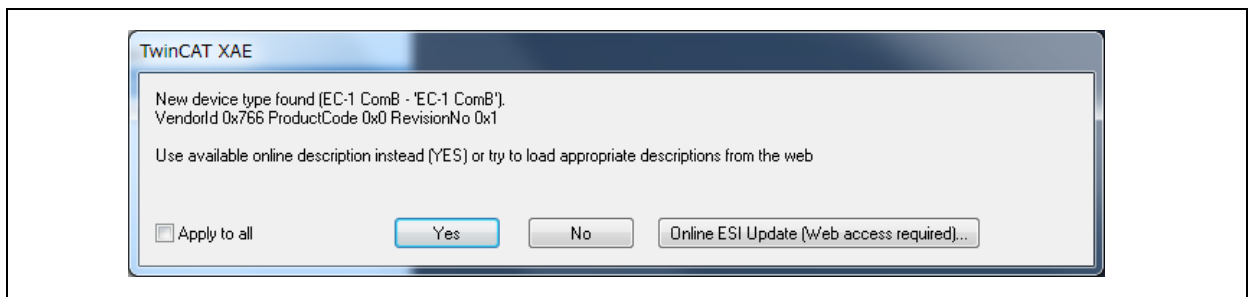
7. スキャンが成功すると接続されている Box (通信ボード) が表示されます。



[確認 1] “Box 1 (PFFFFFFFF RFFFFFFFF)” と表示される場合は、通信ボードの EEPROM へ Slave Information Interface (SII)がプログラムされていないことが考えられます。

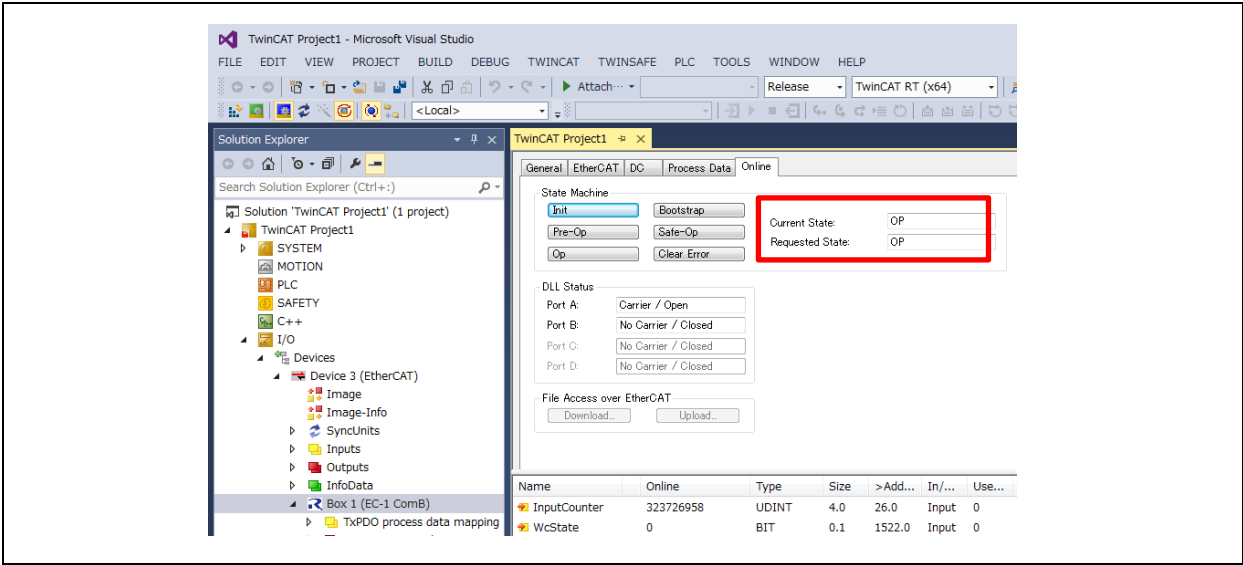
Appendix B を参照して EEPROM をプログラムする必要があります。

[確認 2] 以下の警告が表示される場合は、本項目の手順 1 を参照し ESI ファイルを格納して手順 3 から再度実行してください。



8. 通信状態

EtherCAT 通信が確立していれば、State は OP (Operational)を示します。



4. サンプルアプリケーション操作方法

4.1 EtherCAT サンプルアプリケーション

TwinCAT マスタと通信ボードの TxPDO,RxPDO 通信手法を示します。

EtherCAT 通信ボード用サンプルソフトでは以下のサンプルアプリを実装しています。

TxPDO : InputCounter 4Byte の値を通信ボードから TwinCAT マスタへ送信

RxPDO : OutputCounter 4Byte の値を TwinCAT マスタから通信ボードへ送信

- ・ OutputCounter = 0 : InputCounter は、4Byte インクリメント値を送信
- ・ OutputCounter ≥ 1 : InputCounter は、(OutputCounter + 1) の値を送信

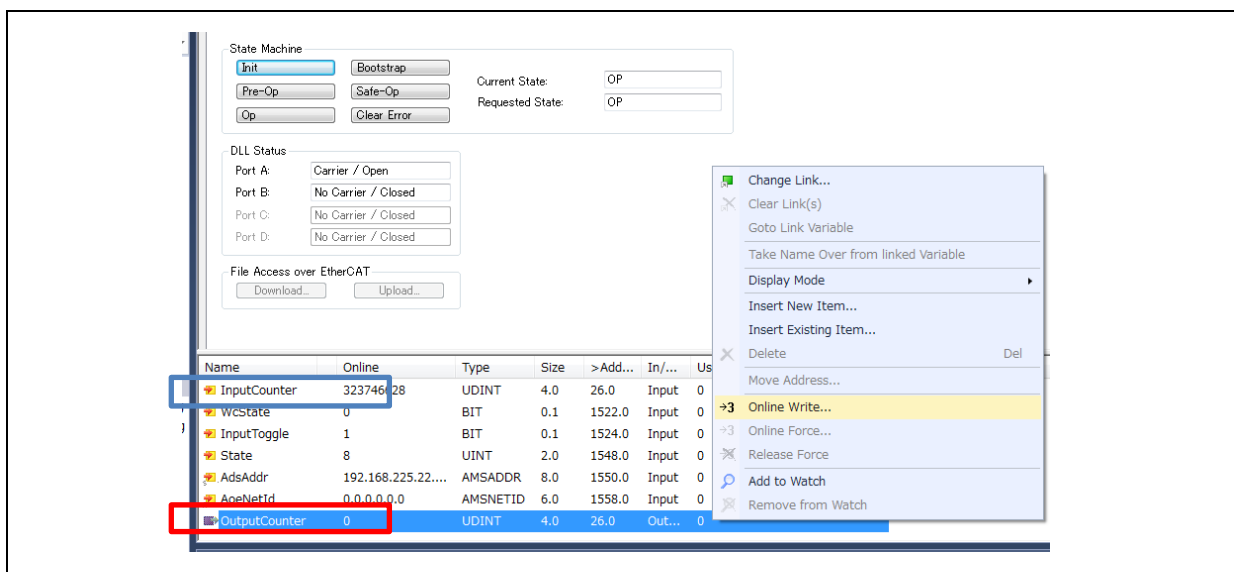
1. InputCounter 読み込みと OutputCounter 書き込み

InputCounter(TxPDO)は、Safe-OP へ遷移するとインクリメントを開始します。

OP へ遷移すると OutputCounter(RxPDO)が有効となります。

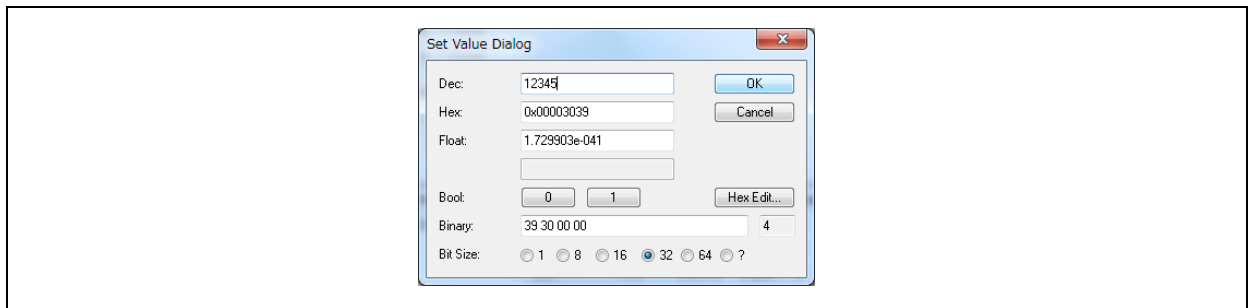
InputCounter は、インクリメント値を読み込みます。

OutputCounter を右クリックし、“Online Write” を選択



2. OutputCounter 値の書き込み

TwinCAT マスタから通信ボードへ送信する RxPDO 4 Byte を入力し、OK を押す



3. InputCounter 値の読み込み

RxPDO 4 Byte 入力値 + 1 の値が、TxPDO の InputCounter として、スレーブから受信する

Name	Online	Type	Size	>Add...	In/...	Use...	Linked to
InputCounter	12346	UDINT	4.0	26.0	Input	0	
WcState	0	BIT	0.1	1522.0	Input	0	
InputToggle	1	BIT	0.1	1524.0	Input	0	
State	8	UINT	2.0	1548.0	Input	0	
AdsAddr	192.168.225.22....	AMSADDR	8.0	1550.0	Input	0	
AoeNetId	0.0.0.0.0.0	AMSNETID	6.0	1558.0	Input	0	
OutputCounter	12345	UDINT	4.0	26.0	Out...	0	

4.2 EtherCAT CiA402 サンプルアプリケーション

TwinCAT マスタと通信ボードの、Cyclic Synchronous Position (CSP) モードの通信手法を示します。

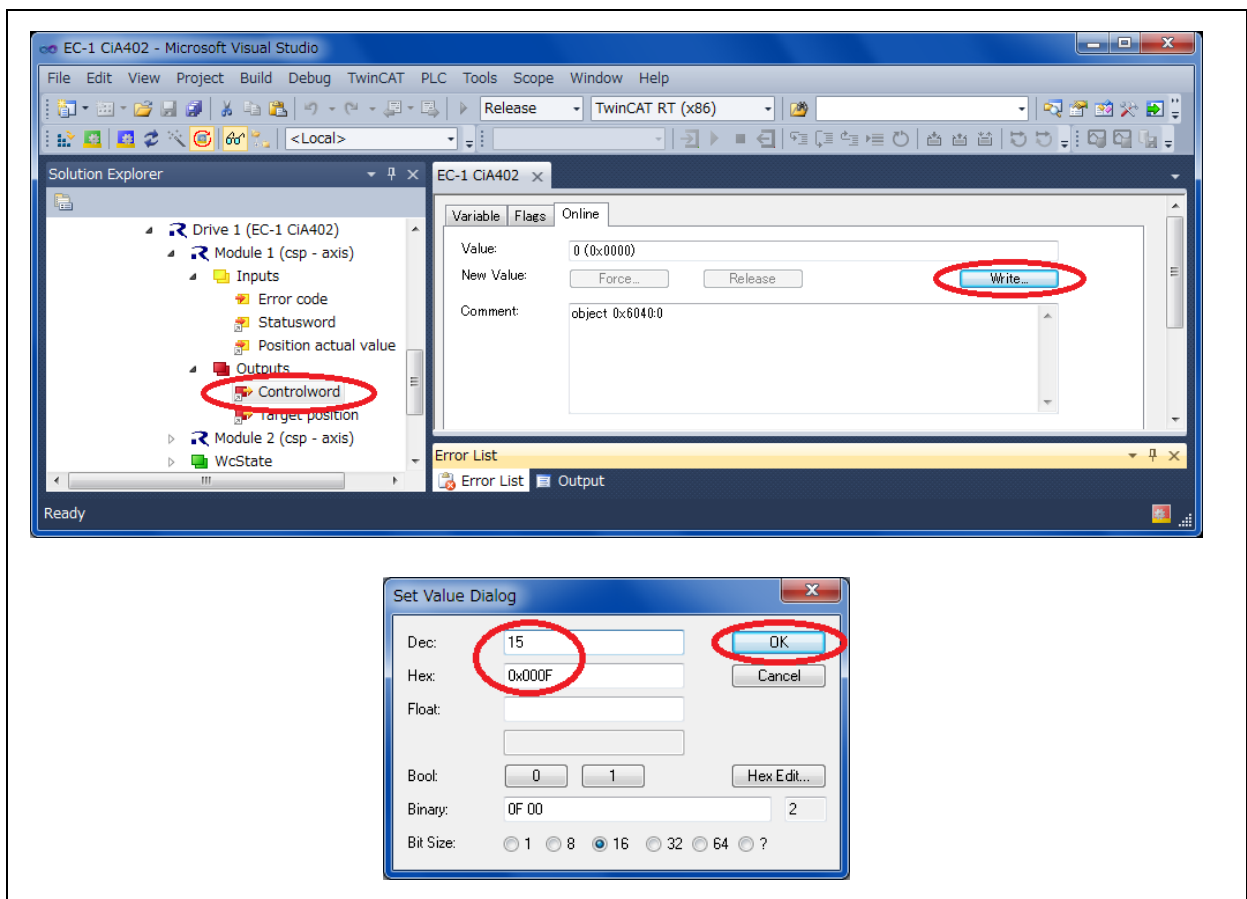
EtherCAT 通信ボード用 CiA402 サンプルソフトでは以下のサンプルアプリを実装しています。

- ・ 1ms タイマ割り込みで、DummyMotor()関数をコール
- ・ DymmyMotor()関数で、Target position(目標位置)と Position actual value(現在位置)とを比較し、目標位置と一致するまで現在位置をインクリメントまたはデクリメント

1. Controlword オブジェクトへの書き込み

Controlword(Outputs)で、CiA402 の状態遷移を制御します。

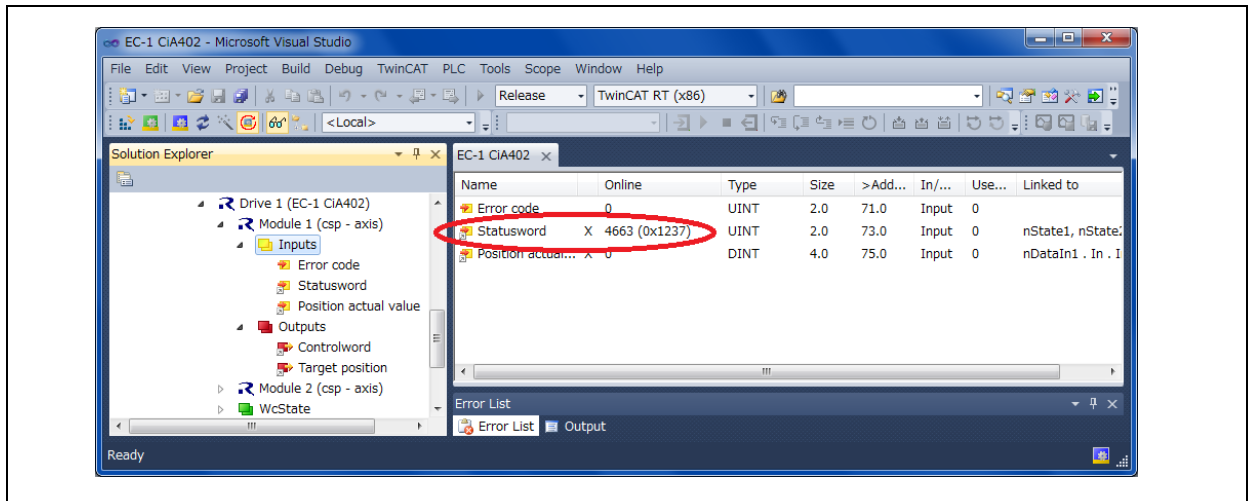
「000Fh(15)」を入力することで、「Operation enabled」の状態になります。



2. Statusword オブジェクトの確認

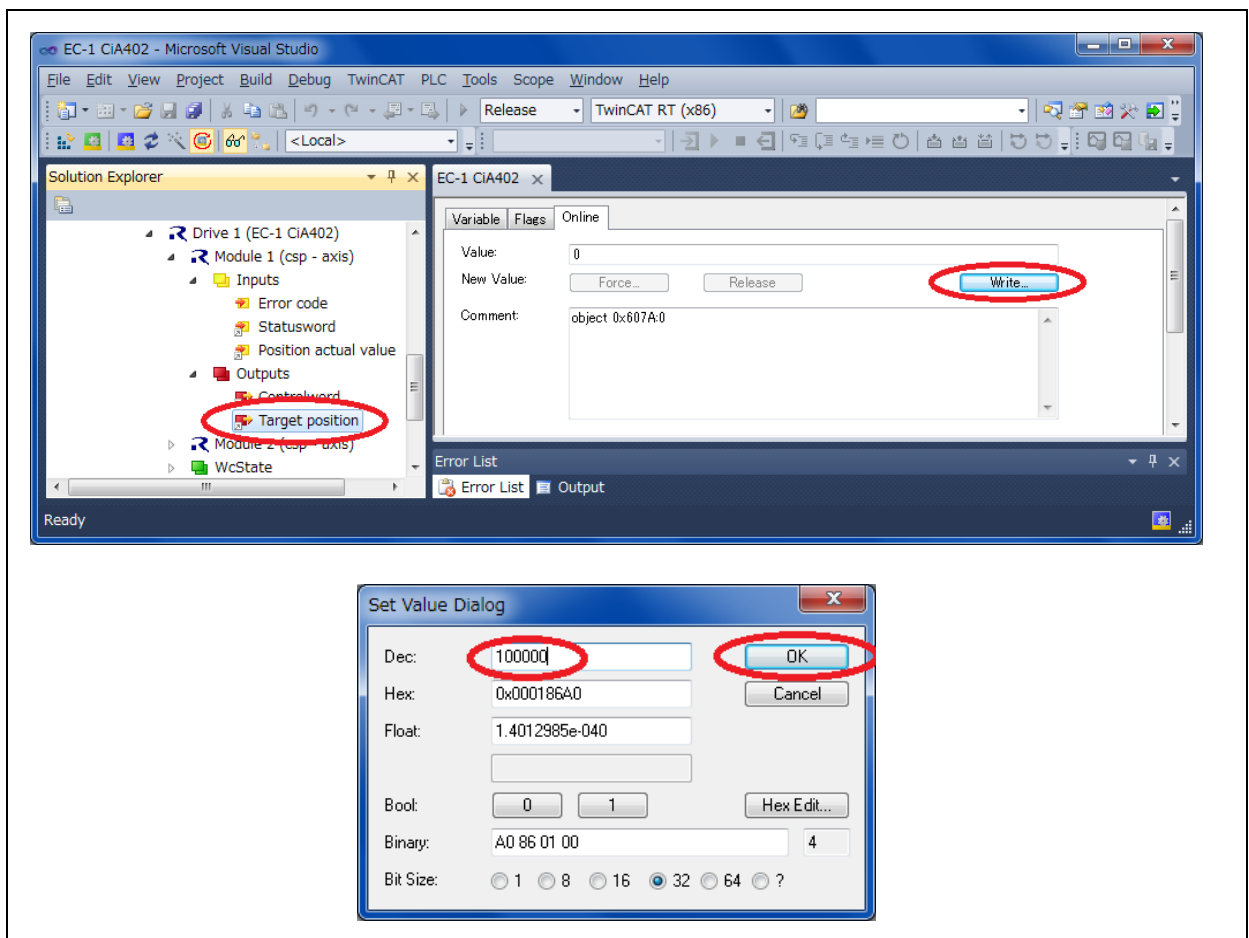
Statusword(Inputs)で、状態の確認をします。

「xxxx xxxx x01x 0111b」が返っていれば正常です。



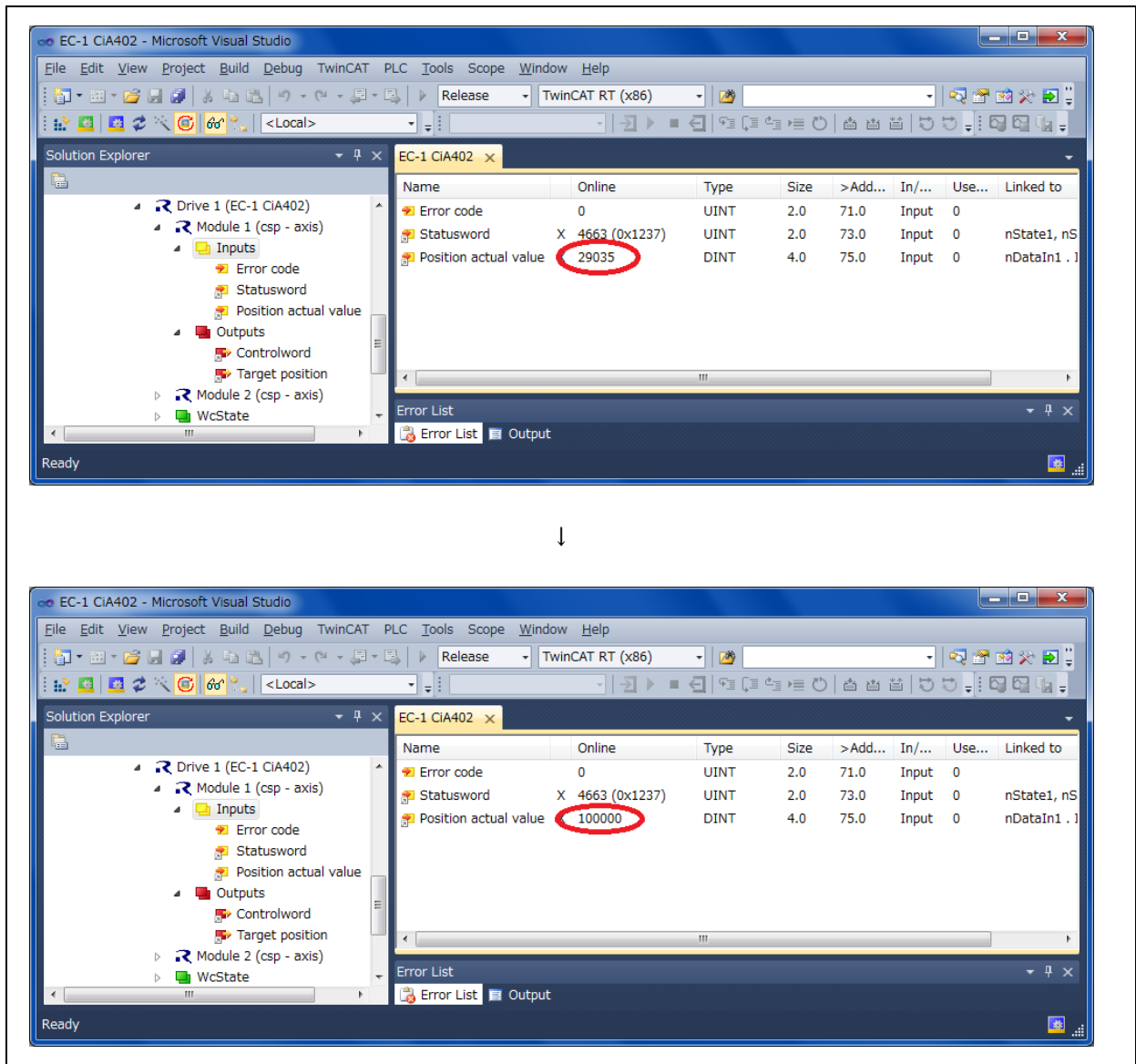
3. Target position オブジェクトへの書き込み

任意の目標位置を入力します。



4. Position actual value オブジェクトの確認

現在位置がインクリメントし、目標位置に到達して停止することを確認します。



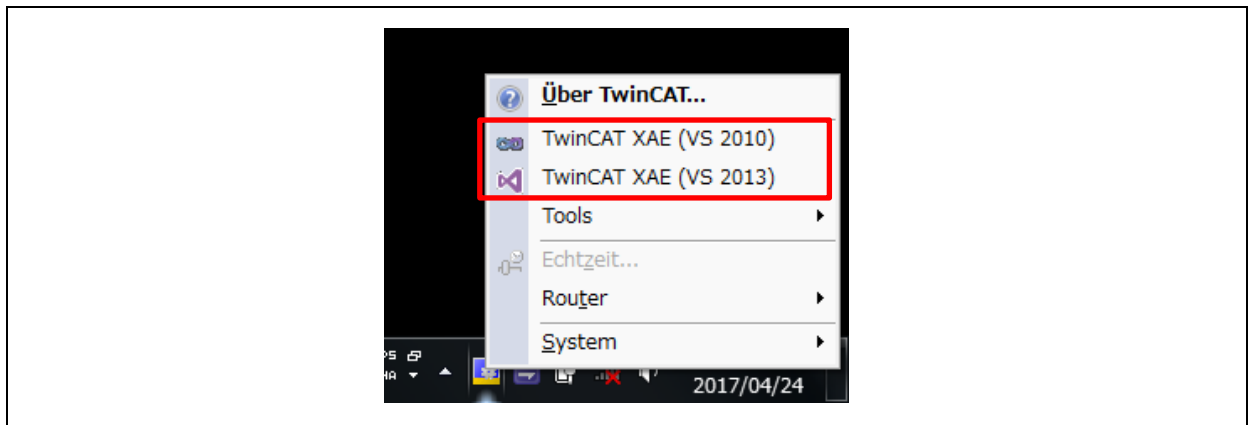
5. Appendix A EtherCAT ドライバ・インストール

TwinCAT を使用するには、TwinCAT ドライバのインストールが必要です。

インストール手順を以下に示します。

1. TwinCAT 起動

タスクから TwinCAT XAE (VS 20xx)をクリック



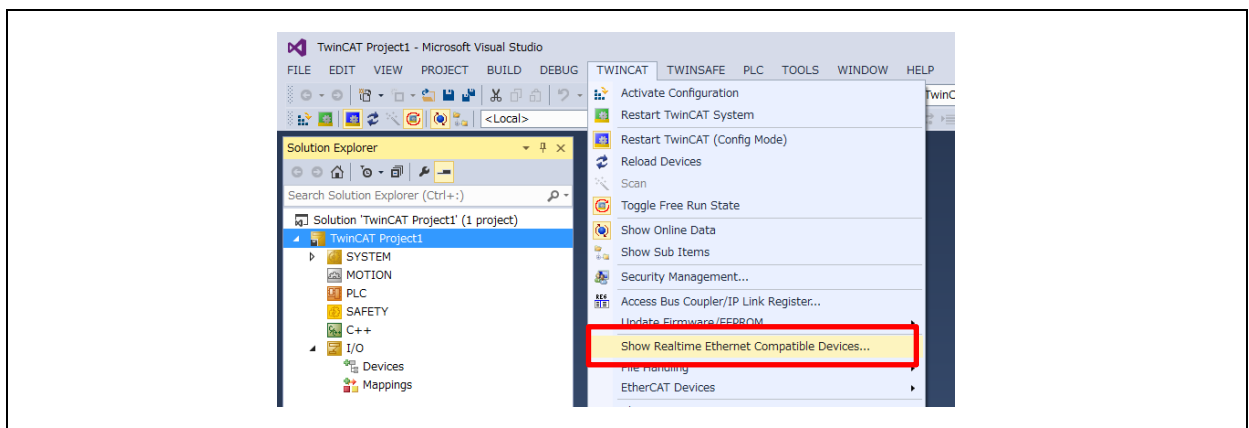
VS 2010, VS2013 どちらでも実行可能です。インストールされている Visual Studio のバージョンのみ表示されます。

2. New Project

New Project を開き TwinCAT を起動します。

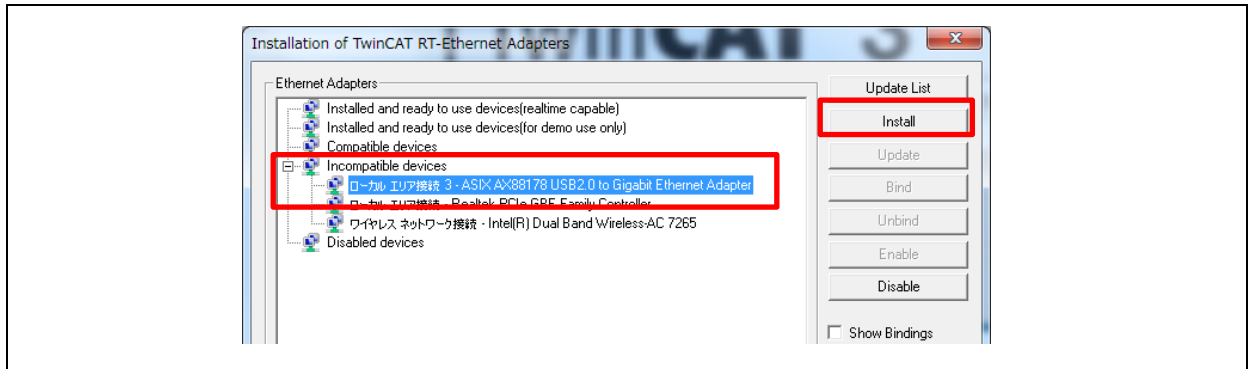
3. イーサネット・アダプタ表示

[TwinCAT] > [Show real Time Ethernet Compatible Devices] を選択

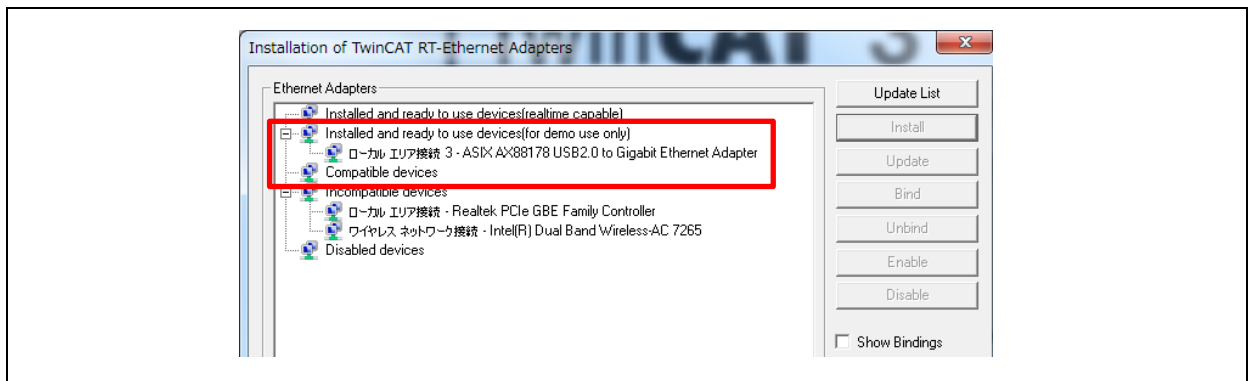


4. TwinCAT ドライバのインストール

インストール先のネットワークアダプタを選択し、[Install] を押します。



インストールが完了すると、[Installed and ready to use devices] にインストールしたネットワークアダプタが表示されれば完了です。



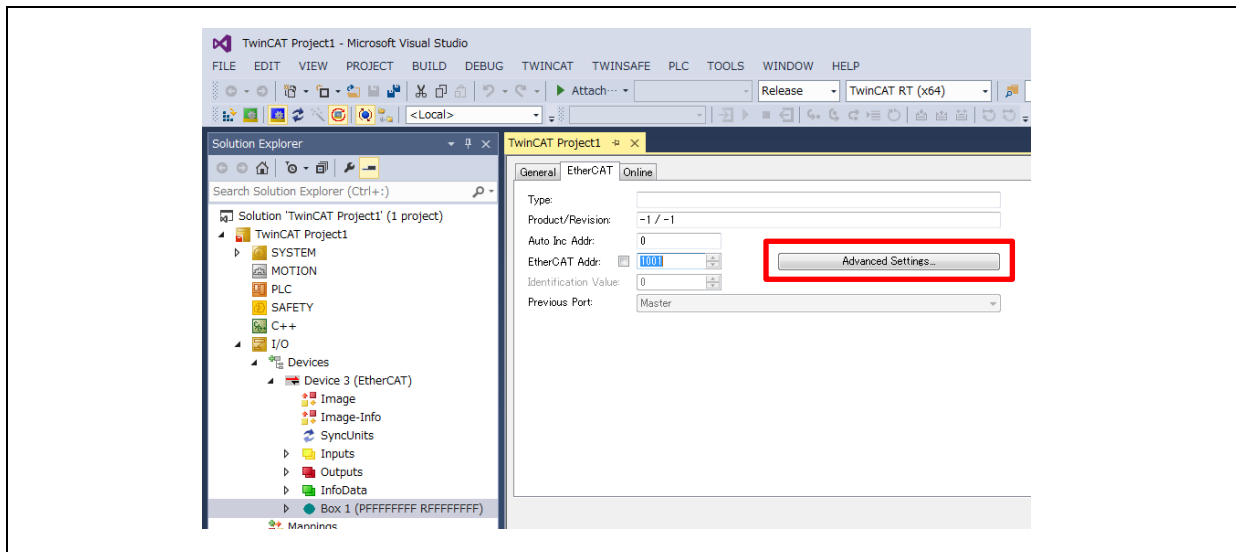
6. Appendix B EEPROM プログラム

EEPROM には、Slave Information Interface (以下、SII)をプログラムされている必要があります。初期状態のボードでは EEPROM がブランクになっているので、下記手順に従い SII をプログラムしてください。

1. Advanced Setting 起動

SII をプログラムするスレーブの Box をダブルクリックし、[EtherCAT]タブを選択。

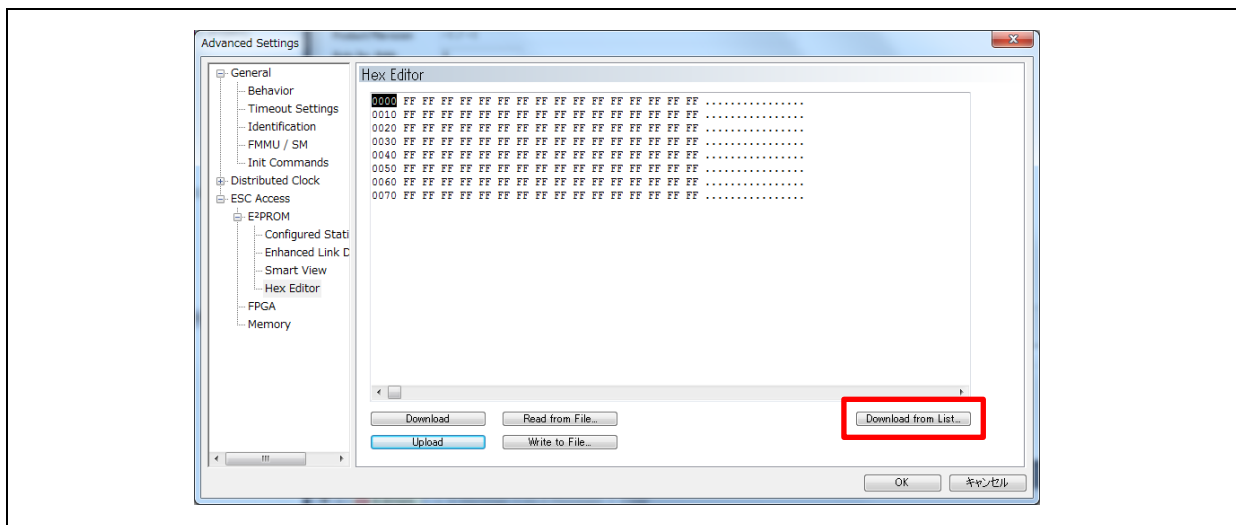
[Advanced Setting] をクリック。



2. Hex Editor

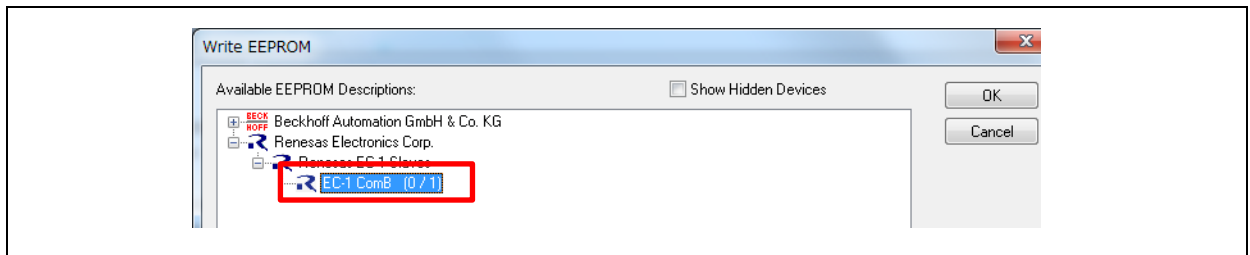
[ESC Access] > [E2PROM] > [Hex Editor] を選択。

[Download From List] をクリック。



3. ESI ファイル選択

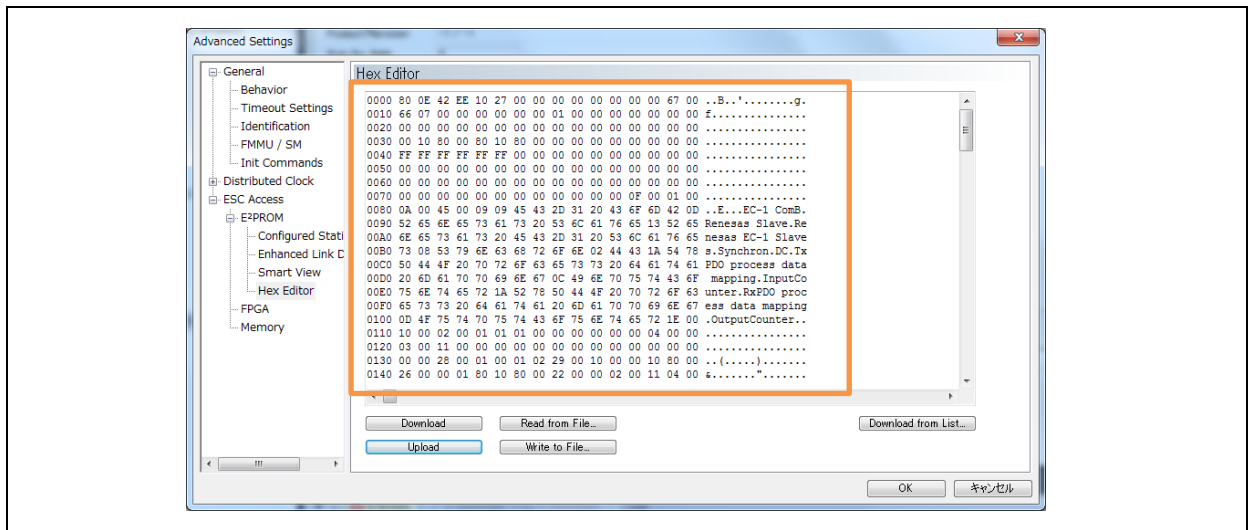
3.3 節の手順 1 で格納した ESI ファイルを選択し、OK を押します。



4. Download

下記図のように Hex 情報が表示されればプログラム完了です。

EEPROM 書き換え後は、ボードの電源をオン、オフしてください。



ホームページとサポート窓口

ルネサス エレクトロニクスホームページ

<http://www.renesas.com/>

お問い合わせ先

<http://www.renesas.com/contact/>

すべての商標および登録商標は、それぞれの所有者に帰属します。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2017.05.10	-	初版発行
1.10	2018.09.04	8	「表 2 5 EtherCAT サンプルアプリケーションのファイル構成」 ファイル構成を更新
		21	「3.3 TwinCAT 接続」 ファイルパスを更新

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。

外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. リザーブアドレス（予約領域）のアクセス禁止

【注意】リザーブアドレス（予約領域）のアクセスを禁止します。

アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。

リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

同じグループのマイコンでも型名が違うと、内部ROM、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

○Arm および Cortex は、Arm Limited（またはその子会社）の EU またはその他の国における登録商標です。 All rights reserved.

○Ethernet およびイーサネットは、富士ゼロックス株式会社の登録商標です。

○IEEE は、the Institute of Electrical and Electronics Engineers, Inc. の登録商標です。

○TRON は” The Real-time Operation system Nucleus” の略称です。

○ITRON は” Industrial TRON” の略称です。

○ μ ITRON は” Micro Industrial TRON” の略称です。

○TRON、ITRON、および μ ITRON は、特定の商品ないし商品群を指す名称ではありません。

○EtherCAT®, および TwinCAT®は、ドイツ Beckhoff Automation GmbH によりライセンスされた特許取得済み技術であり登録商標です。

○その他、本資料中の製品名やサービス名は全てそれぞれの所有者に属する商標または登録商標です。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品、本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。

標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、
家電、工作機械、パーソナル機器、産業用ロボット等

高品質水準： 輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、
金融端末基幹システム、各種安全制御装置等

当社製品は、データシート等により高信頼性、Harsh environment向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。

6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
9. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
10. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
12. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。

注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。

注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。

(Rev.4.0-1 2017.11)



ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：<https://www.renesas.com/contact/>