

Renesas RA ファミリ

クイックスタートガイド : MCK-RA8T2 EtherCAT

要旨

本書は、RA マイコン評価ボードを使用して EtherCAT 通信を評価のためのクイックスタートガイドです。また、TwinCAT®Master 構成ツールを使用してスレーブの動作とスタック機能を確認する手順について説明します。

対象デバイス

MCK-RA8T2

目次

| | |
|---|----|
| 1. 概要..... | 3 |
| 1.1 略語/定義..... | 3 |
| 1.2 参考資料..... | 3 |
| 2. 特徴..... | 4 |
| 3. サンプルアプリケーションの構成..... | 5 |
| 3.1 FSP Drivers..... | 5 |
| 3.2 EtherCAT プロトコルスタック..... | 5 |
| 4. 動作環境..... | 6 |
| 5. ボードの設定と接続..... | 7 |
| 5.1 EtherCAT サブデバイス スタック コードの生成..... | 8 |
| 6. Setting up a TwinCAT3..... | 10 |
| 6.1 Copying the ESI Files..... | 10 |
| 6.2 Add Driver..... | 10 |
| 7. EtherCAT サンプルプログラムの実行..... | 11 |
| 7.1.1 プロジェクトの立ち上げ手順..... | 11 |
| 8. TwinCAT3 への接続..... | 15 |
| 8.1 I/O デバイスのスキャン..... | 15 |
| 8.2 EEPROM データの更新..... | 17 |
| 8.3 I/O 機能の評価..... | 19 |
| 8.4 DC 同期..... | 21 |
| 9. Appendix..... | 23 |
| 9.1 Appendix A: Point of caution when using SSC Tool..... | 23 |
| 9.2 Appendix B : パッチのインストール方法..... | 25 |
| 9.2.1 Using Git for Windows (64bit)..... | 25 |

| | | |
|-------|---|----|
| 9.2.2 | Ushing MinGW Installation Manager | 26 |
| 9.3 | Appendix C : EEPROM 容量設定変更..... | 28 |
| 10. | 制限事項..... | 29 |
| | 改訂記録..... | 30 |

1. 概要

本書は、RA8T2 上で EtherCAT を操作する方法について説明します。

EtherCAT (Ethernet for Control Automation Technology) は、Beckhoff Automation 社が開発した Ethernet ベースのフィールドバスシステムです。EtherCAT の開発は、データ更新時間の短縮、通信ジッタの低減、ハードウェアコストの削減が求められるオートメーションアプリケーション（モーション制御、I/O、センサーなど）に Ethernet を適用することを目的としています。

EtherCAT サブデバイススタックコード生成ツール（SSC ツール）は、ETG 会員に無料で提供されています。ETG ウェブサイトからダウンロードできます。SSC ツールを使用すると、開発者のニーズに合わせてカスタマイズされたスタック、デバイス記述ファイル（ESI）、および個別のソースコードドキュメントを生成できます。

このドキュメントでは、Renesas RA8T2 プラットフォームと互換性のある EtherCAT スタックコードを使用して、EtherCAT サブデバイス機能を評価する手順について説明します。このドキュメントは、SSC ツールを使用して EtherCAT サブデバイススタックコードを生成する方法と、TwinCAT マスターおよびテストアプリケーションに対する動作評価方法を説明することに限定されています。

1.1 略語/定義

表 1.1 略語/定義

| Index | Abbreviations /Definitions | Description |
|-------|----------------------------|---|
| 1 | CiA | CAN in Automation |
| 2 | CoE | CAN application protocol over EtherCAT |
| 3 | DC | Distributed Clocks |
| 4 | EEPROM | Electrically Erasable Programmable Read-Only Memory |
| 5 | ESC | EtherCAT SubDevice Controller |
| 6 | ESI | EtherCAT SubDevice Information |
| 7 | ESM | EtherCAT State Machine |
| 8 | ETG | EtherCAT Technology Group |
| 9 | EtherCAT | Ethernet for Control Automation Technology |
| 10 | I/O | Input/Output |
| 11 | OD | Object Dictionary |
| 12 | PDO | Process Data Object |
| 13 | SDO | Service Data Objects |

1.2 参考資料

EtherCAT に関する技術情報は ETG 組織サイトから入手でき、MCK-RA8T2 に関する情報はルネサスから入手できます。

表 1.2 技術資料

| Index | Technical Inputs |
|-------|---------------------------|
| 1 | r01ds0436xx01xx-ra8t2.pdf |
| 2 | r01uh1067xx01xx-ra8t2.pdf |
| | |

2. 特徴

SSC ツールによって生成された EtherCAT スレーブスタックコードは、EtherCAT スレーブコントローラの機能を提供します。

以下の機能が含まれます。

- ESM (EtherCAT State Machine)
- Mailbox protocols:
 - CoE (CAN application protocol over EtherCAT)

- Synchronization Modes:
 - Free Run
 - Sync Manager Synchronization
 - DC Synchronization

- I/O function:
 - I/O Input DIP SW
 - I/O Output LED



EtherCAT is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.

EtherCAT の詳細については、以下のサイトをご覧ください。

[EtherCAT Technology Group | HOME](#)

注)、バージョンはアップデートによって異なる場合があります。最新のマニュアルを参照してください。

3. サンプルアプリケーションの構成

このサンプル アプリケーションは 3 つのブロックで構成されています

- FSP ドライバ
- SSC ツールを使用した EtherCAT プロトコルスタックのサンプルプログラム。

3.1 FSP Drivers

- e² studio/ra
このディレクトリには、EtherCAT プロトコル スタックに必要なコードが含まれています。
 - BSP (Board Support Package) for MCK-RA8T2
 - HAL (Hardware Abstraction Layer) drivers
 - SSC ボーディングモジュール
- e² studio/ra_gen
ra_genフォルダには、RA構成エディタによって生成されたコードが含まれています。これにはジュールの制御構造と構成構造のグローバル変数が含まれます。
- e² studio/ra_cfg
ra_cfg フォルダには、各モジュールの設定ヘッダーファイルが格納されます。これらのヘッダーファイルで提供される内容については、FSP ビルド時設定を参照してください。

3.2 EtherCAT プロトコルスタック

- e² studio/src/ethercat/beckhoff
SSCツールで生成したソースファイルをコピーするフォルダです。
EtherCATプロトコルスタックが実装されます。
- e² studio/src/ethercat/renesas
MCK の DIPSW と LED を制御するアプリケーションです。

4. 動作環境

本書のサンプルプログラムは以下の環境で動作します。

表 4.1 動作環境

| Item | Description |
|------------------------------------|---|
| Board | MCK-RA8T2 |
| CPU | Cortex [®] -M85 (CPU0) Cortex [®] -M33 (CPU1) |
| Operating frequency | CPU clock (CPUCLK): <ul style="list-style-type: none">• 1 GHz (Cortex[®]-M85)• 250 MHz (Cortex[®]-M33) |
| Operating voltage | 3.3 V |
| Device requirements | R7KA8T2LFECAC <ul style="list-style-type: none">• MRAM Capacity: 1 MB• SRAM Capacity: 2 MB |
| Integrated development environment | e ² studio 2025-07 or later |
| MCU software package | FSP (Flexible Software Package) v6.1.0 or later |
| Emulator | J-LINK OB |
| Communications protocol | EtherCAT |
| Master tool | TwinCAT3 |

5. ボードの設定と接続

以下に示すように、PC を MCK-RA8T2 ボードに接続します。
 電源は USB-C ケーブルをボードに接続することで供給されます。
 Ethercat 通信の場合は RJ45 コネクタを使用し、LAN ケーブルで PC に接続します。

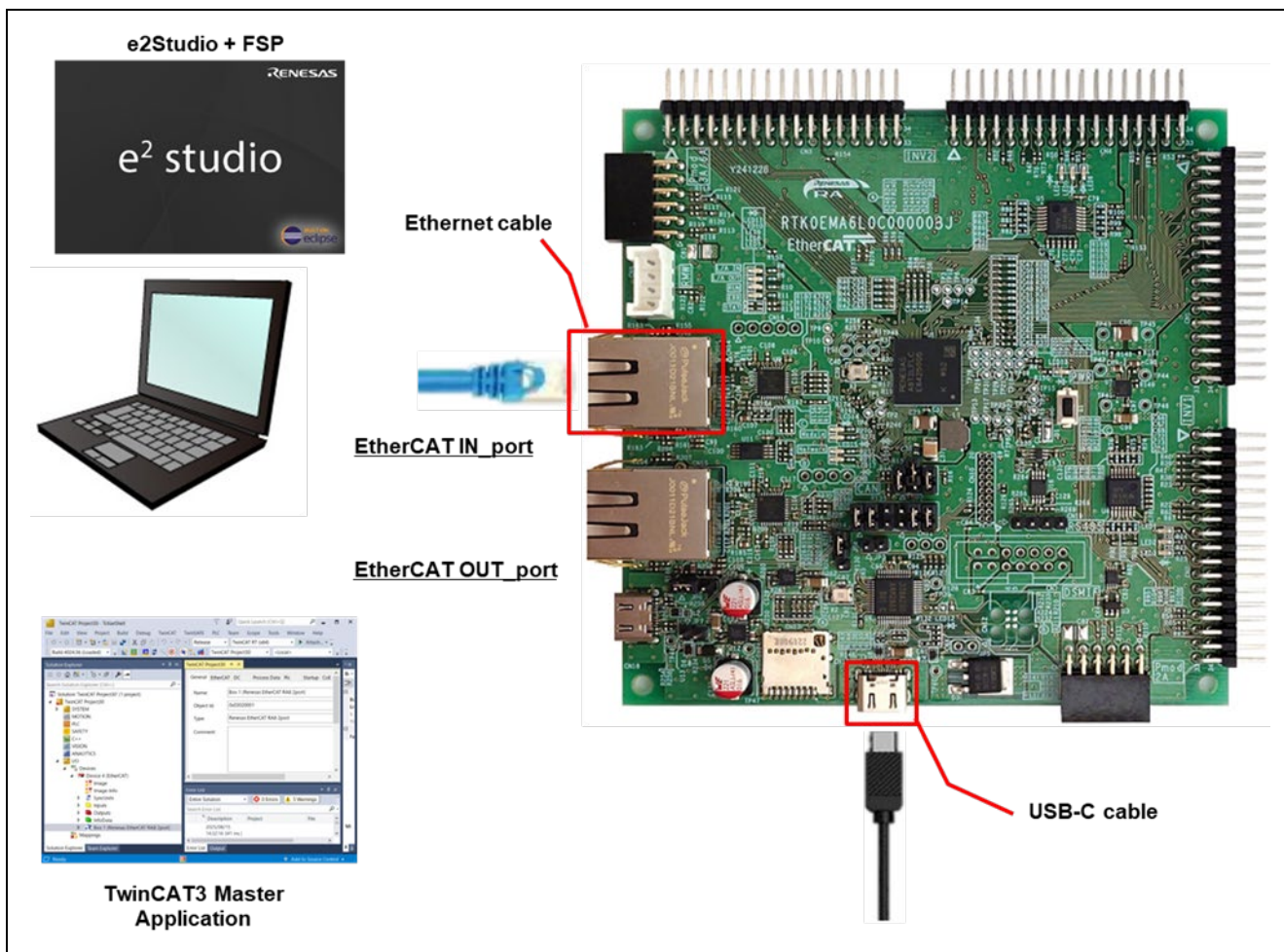


図 1. MCK-RA8T2 board 接続設定

5.1 EtherCAT サブデバイス スタック コードの生成

この章では、SSC ツールを使用して EtherCAT サブデバイススタックコードを生成する方法について説明します。

- * EtherCAT スレーブスタックコードは、このサンプルプロジェクトには含まれていません。
- * EtherCAT スレーブスタックコードを生成するには、EtherCAT スレーブスタックコード(SSC)ツールが必要です。
- * SSC ツールは ETG 協会から入手できます。

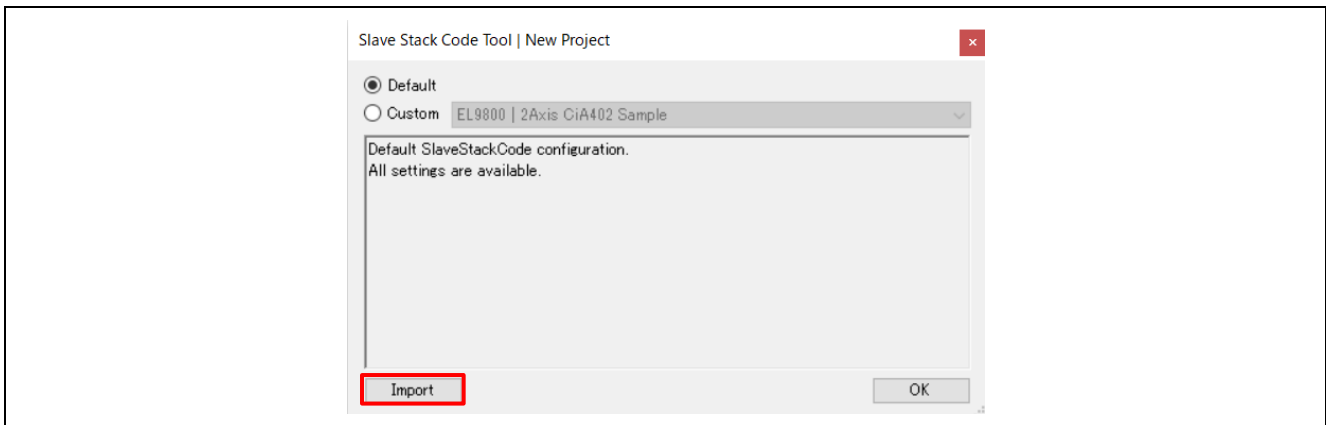
[EtherCAT Technology Group | EtherCAT Slave Stack Code \(SSC\) ET9300](#)

注)、SSC Tool を使用する際の注意点があります。

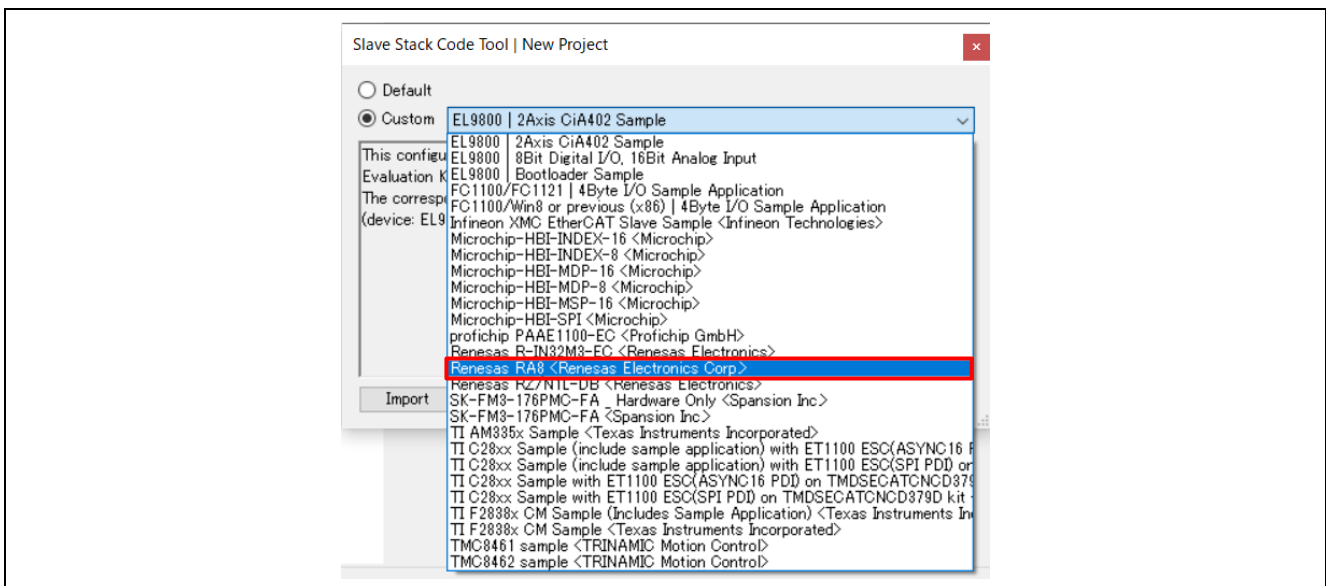
事前に「付録 A:SSC ツールを使用する注意点」を参照してください。

1. Windows のスタートメニューから[EtherCAT Slave Stack Code]→[SSC Tool]をクリックして、SSC ツールを起動します。
2. 新規プロジェクトダイアログでインポートをクリックし、サンプルプログラムフォルダから次の SSC ツール設定ファイルを選択して **OK** をクリックします。

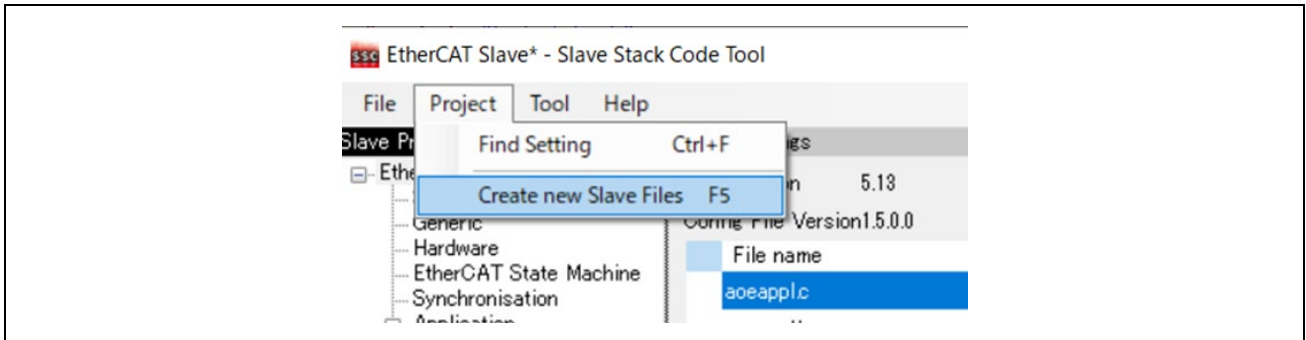
"RA8T2_EtherCAT_MCK_rev0100¥common¥ecat_IO¥SSCconfig¥Renesas_RA8_config.xml"



3. [Custom]にチェックを入れ、一覧から"**Renesas RA8 <Renesas Electronics Corp>**"を選択し、**OK** をクリックします。

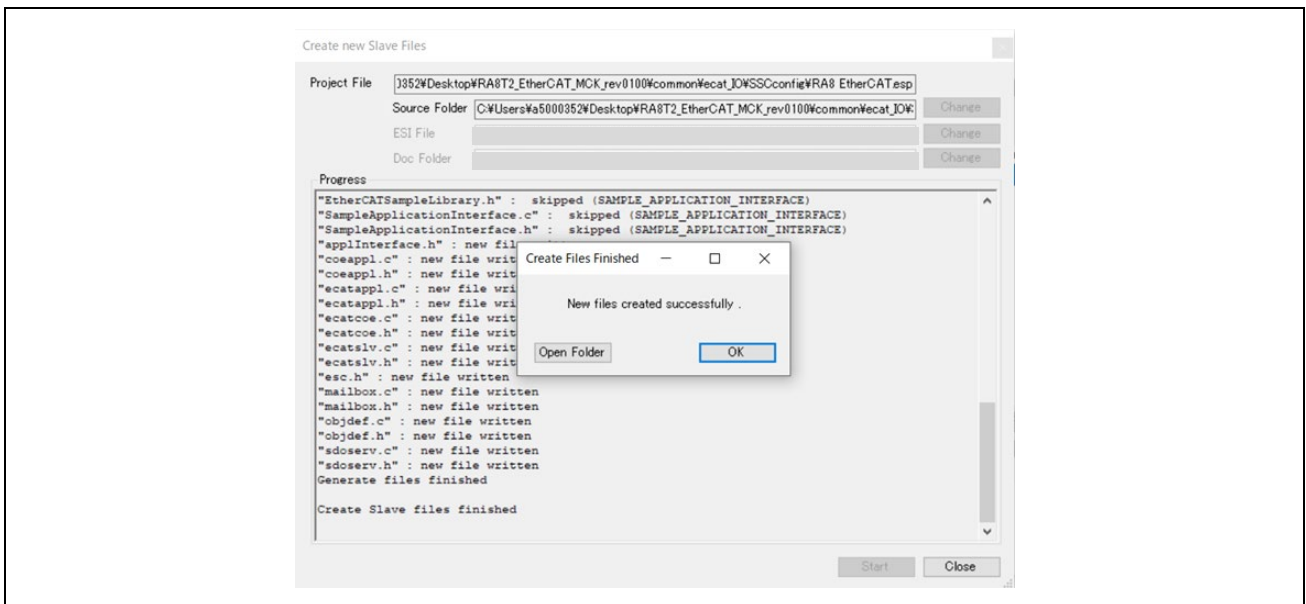


4. [Project]をクリック → [Create New Slave Files]をクリックします。[Current new Slave Files]ダイアログの[Start]をクリックします。



5. ソースコードが生成され、成功すると「New Files created successfully」と表示されるので、[OK]をクリックします。作成処理は完了し、ソースフォルダ「Src」は以下のフォルダにあります。

“RA8T2_EtherCAT_MCK_rev0100¥common¥ecat_IO¥SSCconfig”



6. 生成された EtherCAT スレーブスタックコードを EtherCAT アプリケーションソースフォルダに移動します。
sampleappl.c と sampleappl.h は、“¥application¥ecat” フォルダに格納されます。
スレーブスタックコードを applicaiotn フォルダに移動する際は、これらのファイルを削除しないように注意してください。
フォルダから Src コードを削除するか、e² studio のビルドターゲットから Src コードを除外します。

ソースフォルダ:

RA8T2_EtherCAT_MCK_rev0100¥common¥ecat_IO¥SSCconfig¥Src

移動先フォルダ:

**RA8T2_EtherCAT_MCK_rev0100¥project¥ra8t2_mck_ra8t2¥ecat_IO¥e2studio
¥src¥ethercat¥beckhoff**

6. Setting up a TwinCAT3

6.1 Copying the ESI Files

TwinCAT を起動する前に、リリースフォルダに含まれる ESI ファイル(.xml)を以下の TwinCAT フォルダにコピーします。

リリースフォルダ:

“RA8T2_EtherCAT_MCK_rev0100¥common¥ecat_IO¥ESI”

ESI ファイル名:

“Renesas EtherCAT RA8.xml”

TwinCAT 格納フォルダ:

“\TwinCAT\3.x\Config\IO\EtherCAT”

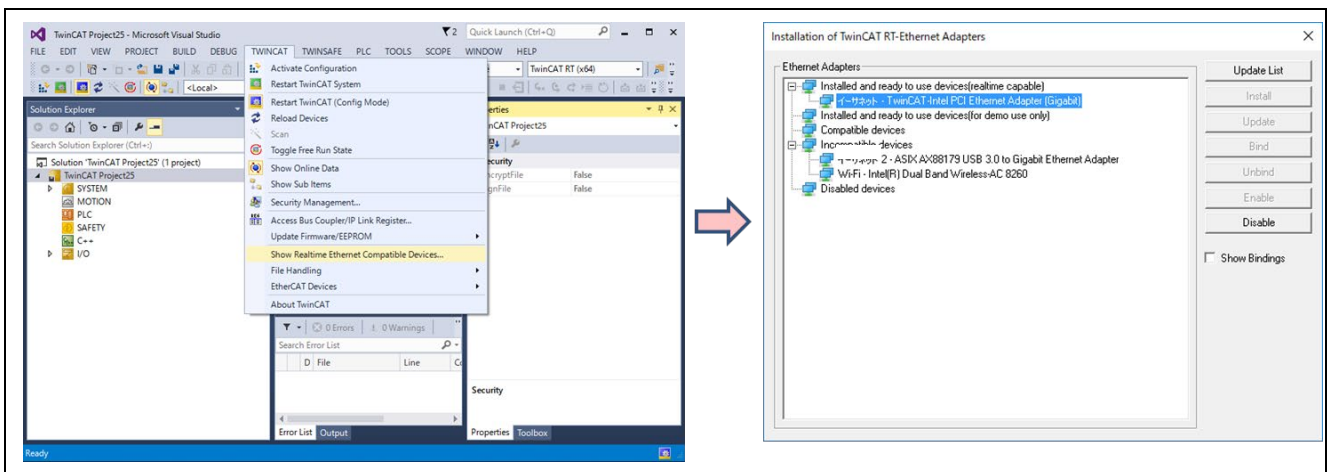
6.2 Add Driver

TwinCAT 用の Ether ドライバを追加します。(初回のみ)

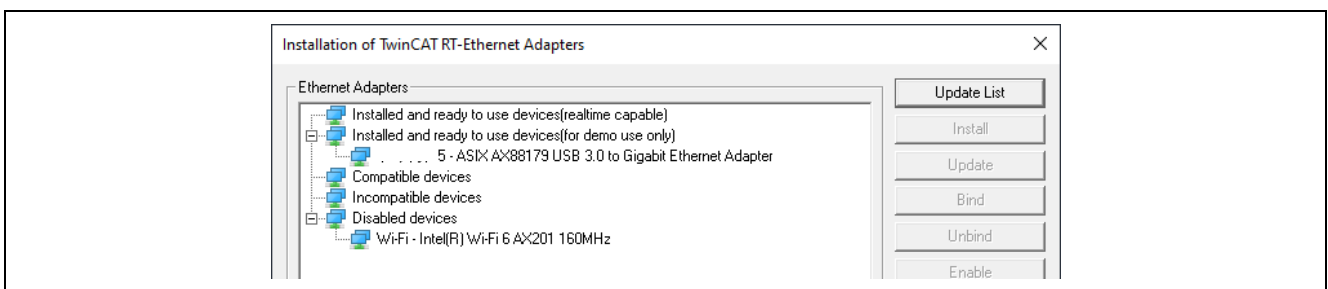
スタートメニューから[TwinCAT3]→[Show Realtime Ethernet Compatible Devices...]を選択。

通信ポートから接続された Ether ポートを選択しインストールします。

選択したイーサネットアダプタが[Installed and ready to use devices(realtime capable)]に移動していれば問題ありません。



注)、Intel 製の NIC を使用しない場合、イーサネットアダプタは [Installed and ready to use devices(realtime capable)]に移動します。



7. EtherCAT サンプルプログラムの実行

この章では、EtherCAT サンプルアプリケーションとの通信手順について説明します。

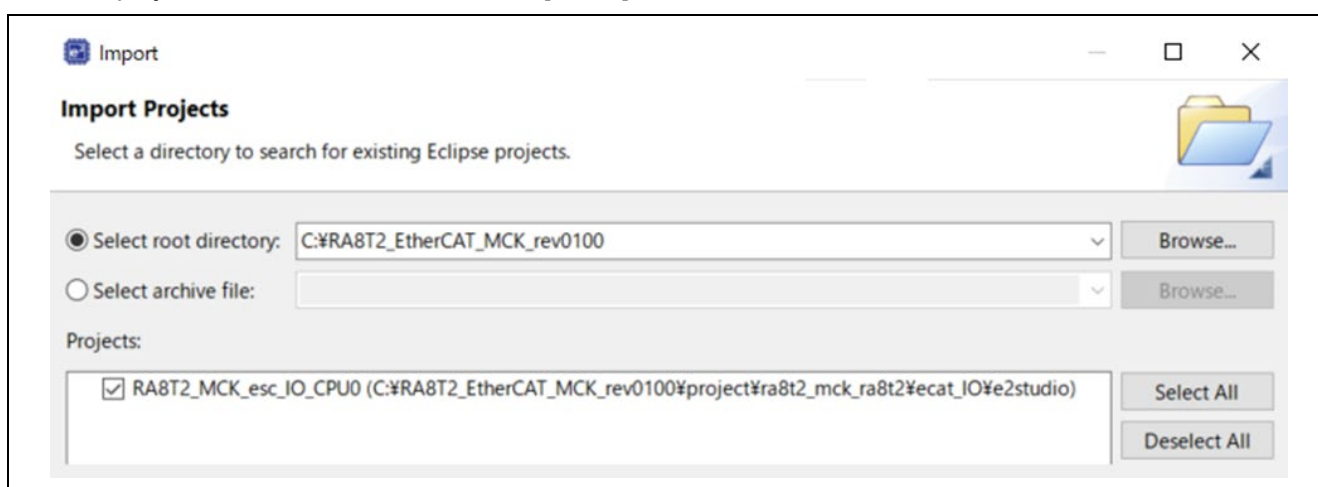
事前にセクション4「動作環境」およびセクション5「ボードの設定と接続」を参照して、ツールのインストールとハードウェア接続を完了してください。

サンプルコードをビルドし、ルネサスエレクトロニクスの e² studio を使用してロードします。

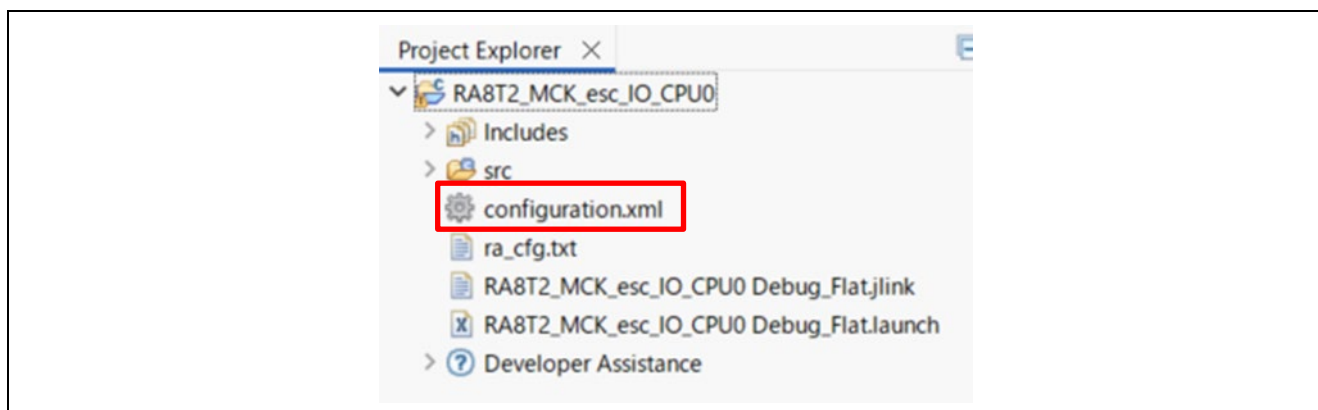
注)、e² studio をインストールし、事前に FSP Pack v6.1.0 を適応させてください

7.1.1 プロジェクトの立ち上げ手順

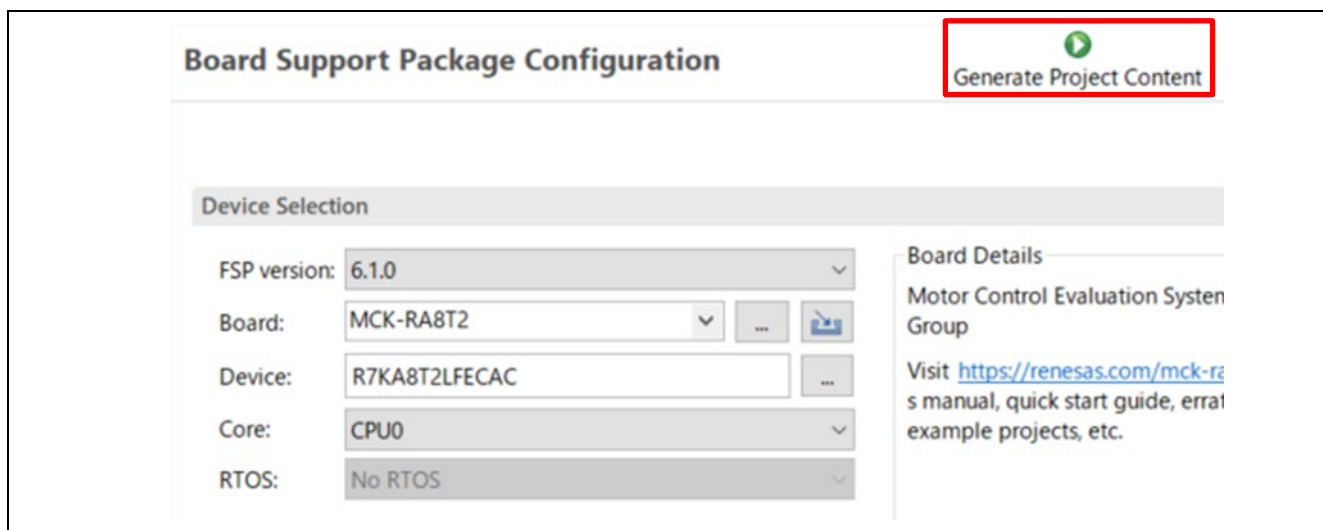
1. サンプルプロジェクトをインポートします。プログラムを起動したら、[File]→[Import]→[Existing Projects into Workspace]を選択します。「Select root directory」にチェックを入れます、「**RA8T2_EtherCAT_MCK_rev0100**」フォルダを選択、I/O project にチェックを入れます。→ [Finish].



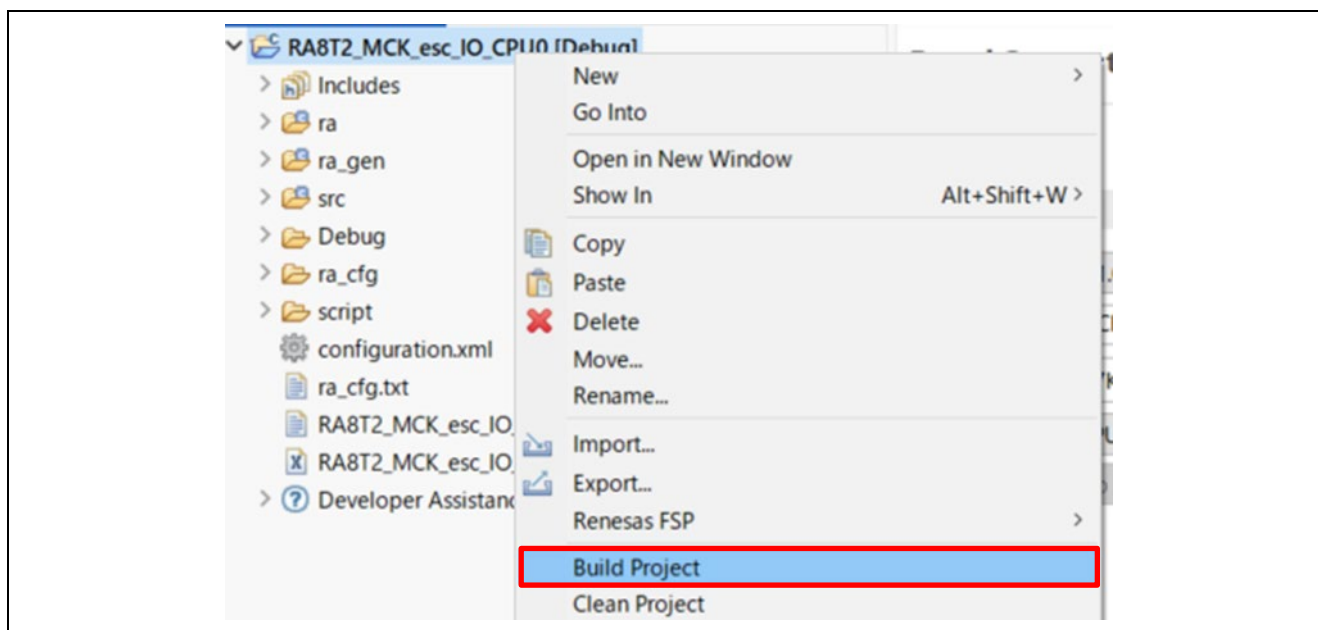
2. 「RA8T2_MCK_esc_IO_CPU0」プロジェクトで「configuration.xml」を開きます。



3. 「Generate Project Content」でコードを生成します。

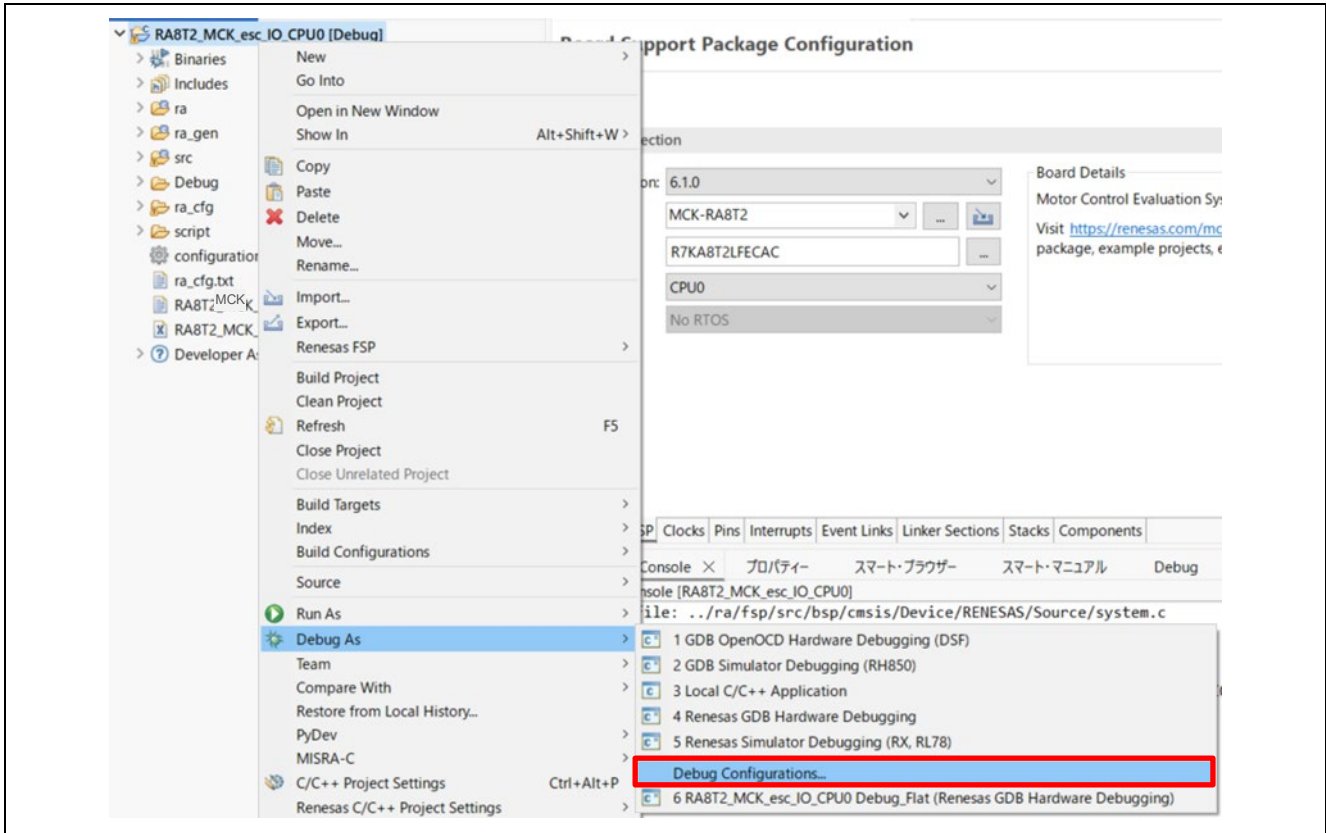


4. ターゲットプロジェクトを選択し、ビルドを実行します。

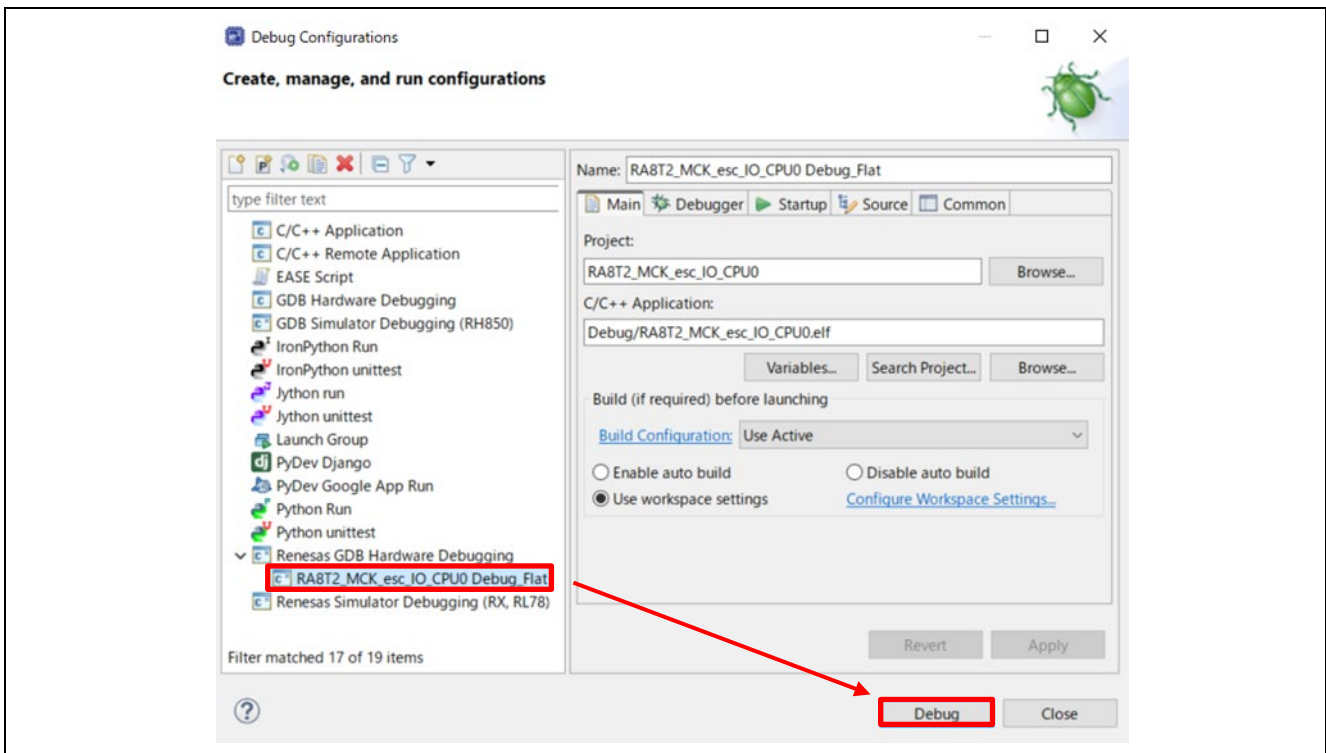


5. RA8T2 MCK ボードの「RESET」スイッチを押します。

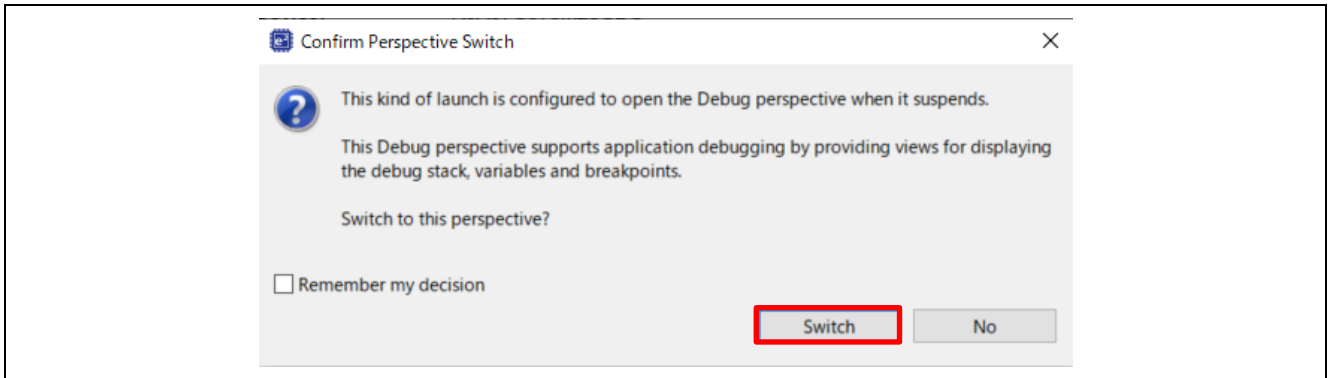
6. ボードと J-Link を接続したら、以下の手順でデバッグを開始します。
 [Project Explorer]画面にて、デバッグするプロジェクトのノードを右クリックし、[Debug As]→[Debug Configurations]を選択します。



[Renesas DBG Hardware Debugging] → [RA8T2_MCK_esc_IO_CPU0 Debug_Flat], 次に、[Debug]を
 押します。

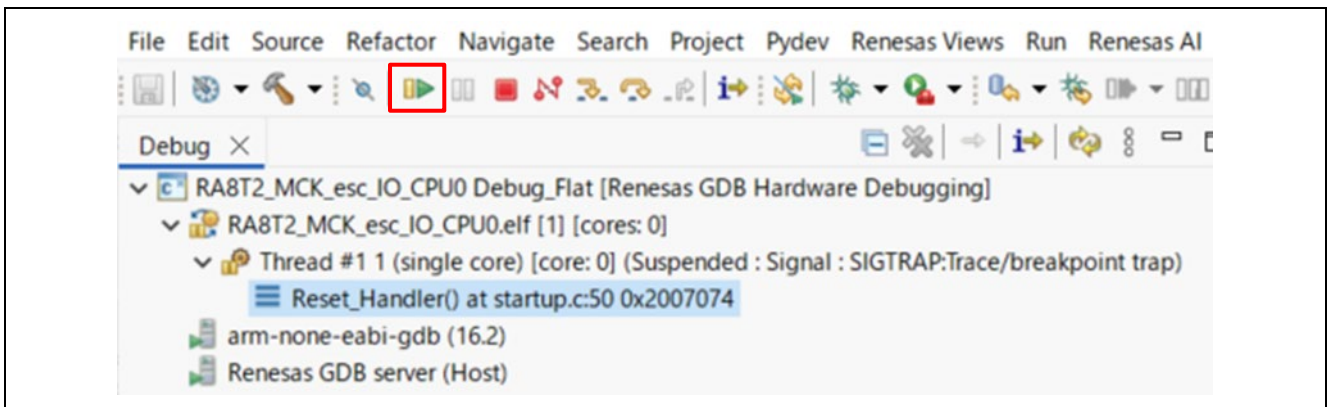


以下のダイアログが表示されるので、デバッグ画面に切り替わります。

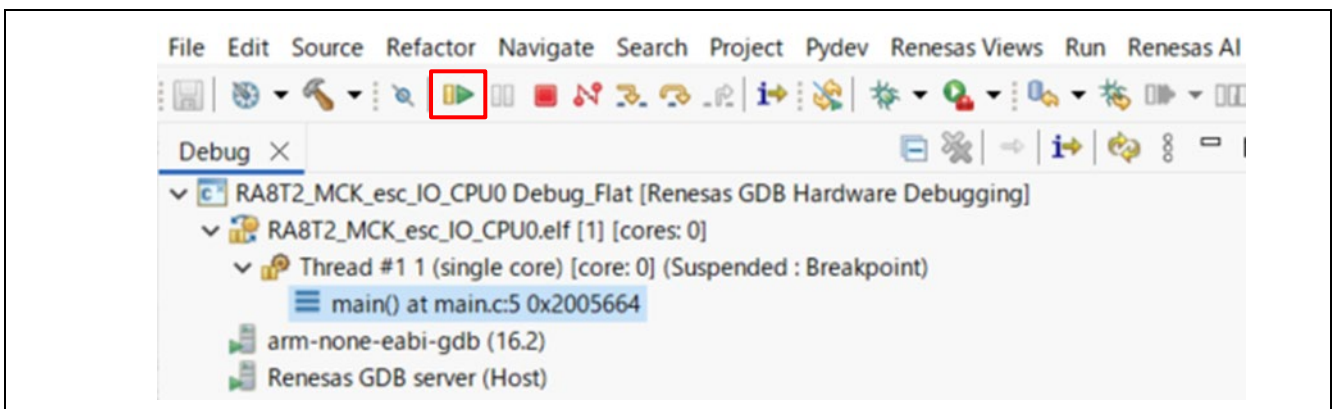


7. 「再開」ボタンを押します。

デバッグを開始すると、main.c の "hal_entry();" で プログラムが中断されます。



もう一度「再開」ボタンを押します。プログラムが実行されます。



8. TwinCAT3 への接続

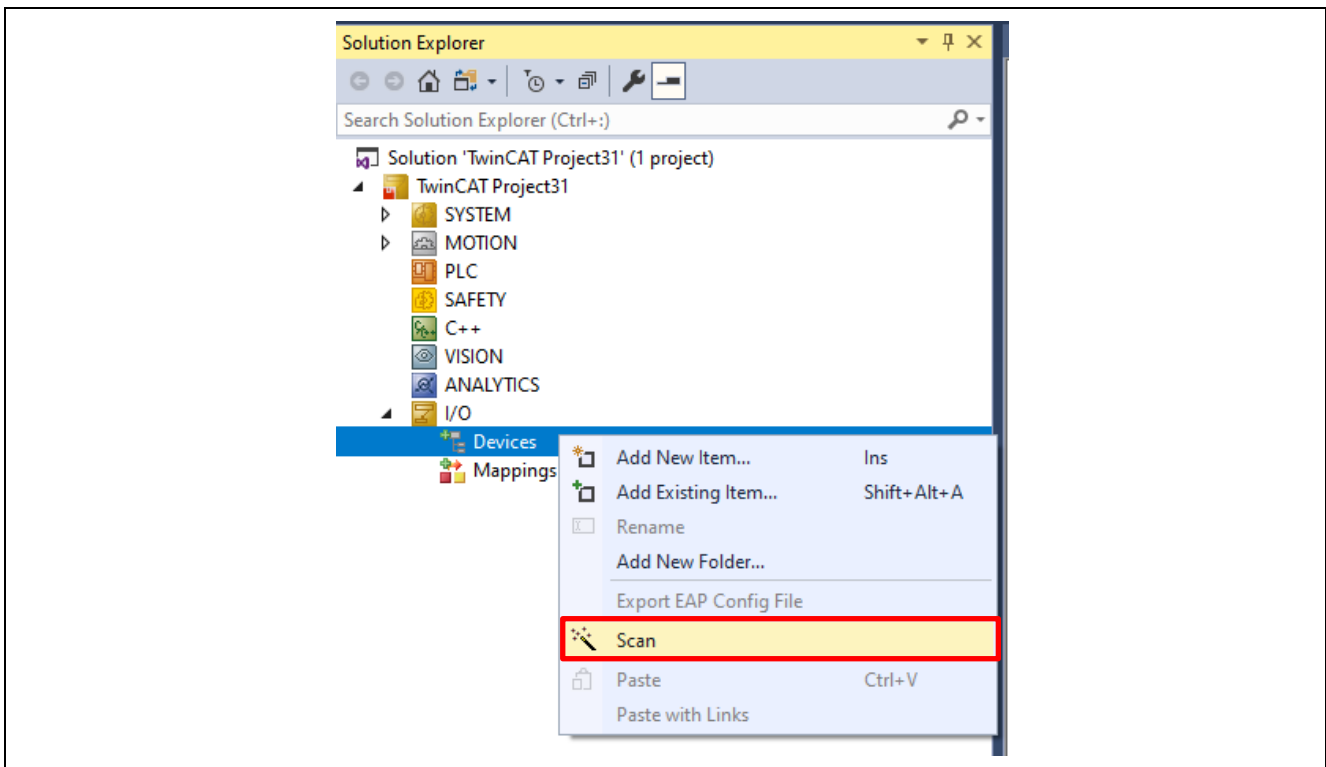
以下で説明する手順を使用して、TwinCAT3 を起動します。

スタートメニューから[Beckhoff]→[TwinCAT3]→[TwinCAT XAE Shell]を選択します。

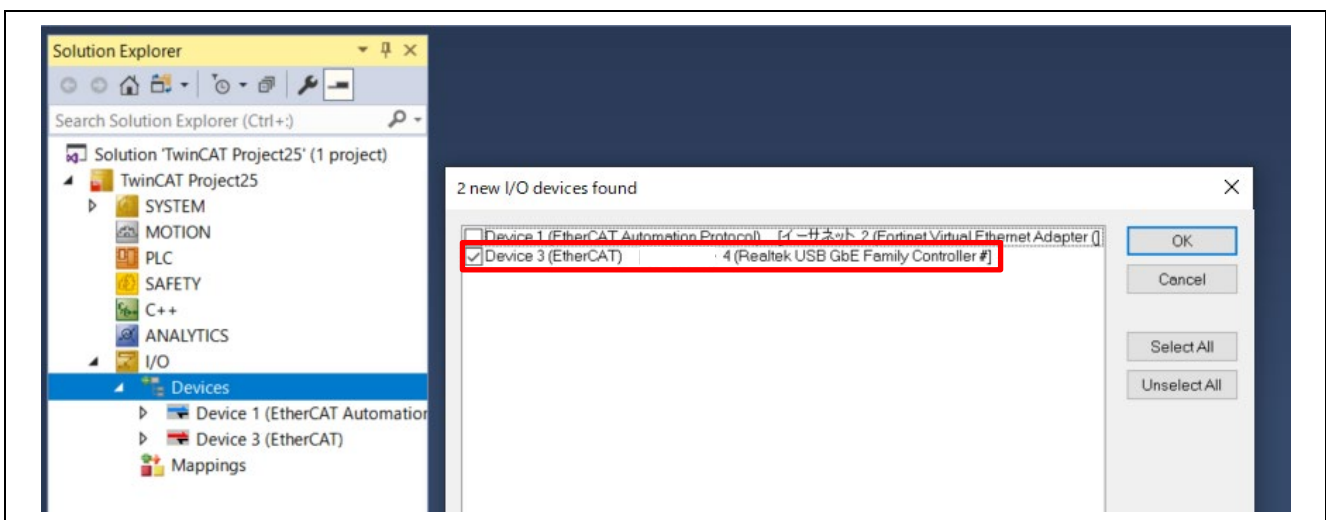
プログラム起動後、[ファイル]→[新規]→[プロジェクト]を選択して、TwinCAT XAE プロジェクトタイプのプロジェクトを新規作成します。その後の手順を以下に説明します。

8.1 I/O デバイスのスキャン

1. (Scan for devices): solution explorer -> I/O -> Devicesで、以下のように「Scan」を選択します。

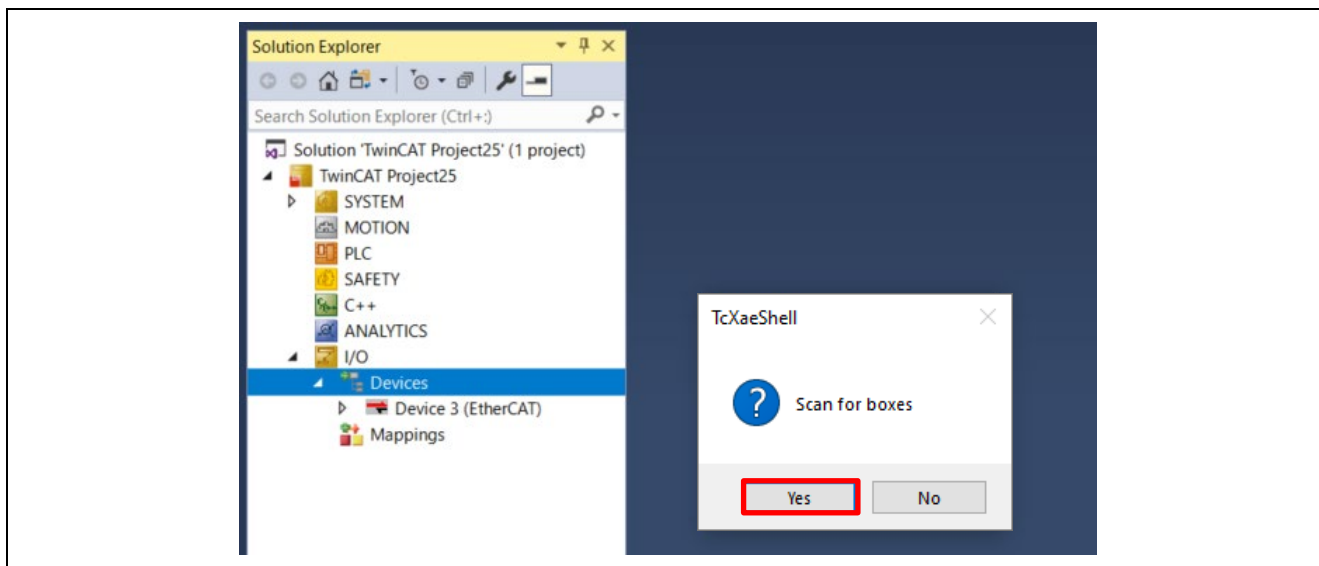


2. (Selecting port):EtherCATポートが以下のように表示されます。選択して[OK]を押します。

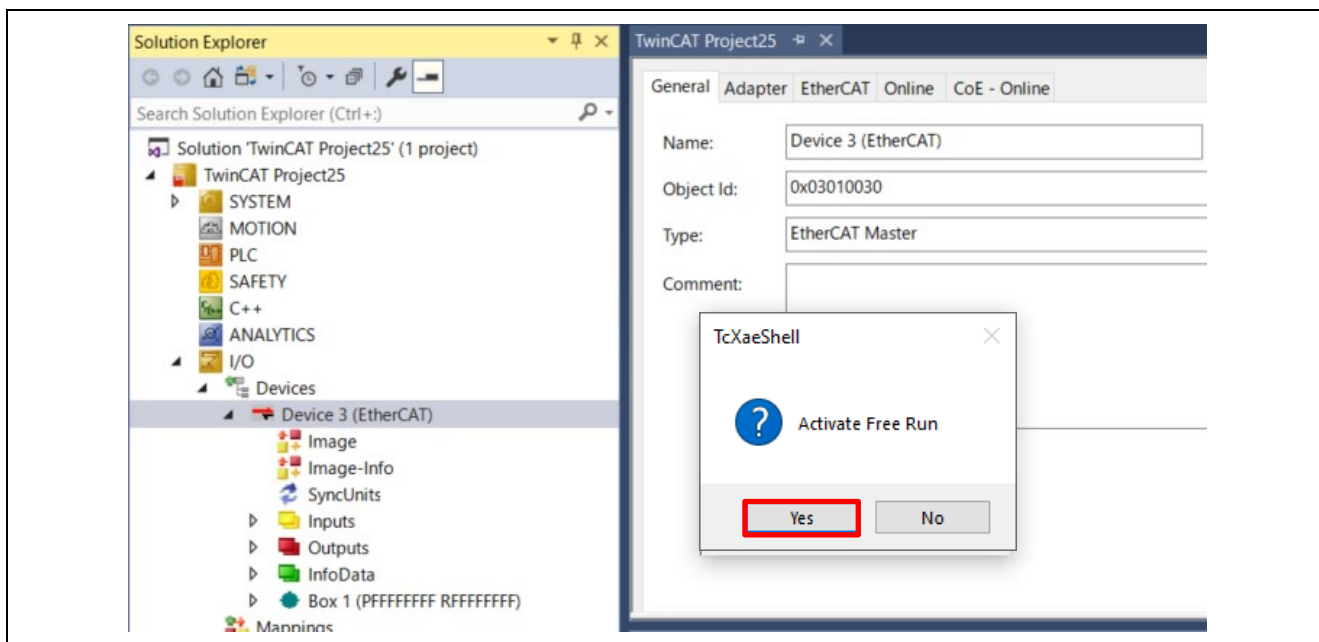


注)、有効な EtherCAT サブデバイスがネットワークに存在する場合、TwinCAT はチェックボックス付きの候補を表示します。

3. デバイスのスキャンを開始する。



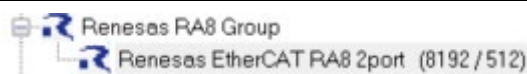
4. (Activate SubDevice): サブデバイスがボックスにリストされます。[Activate free run]ダイアログボックスの[Yes]をクリックします。



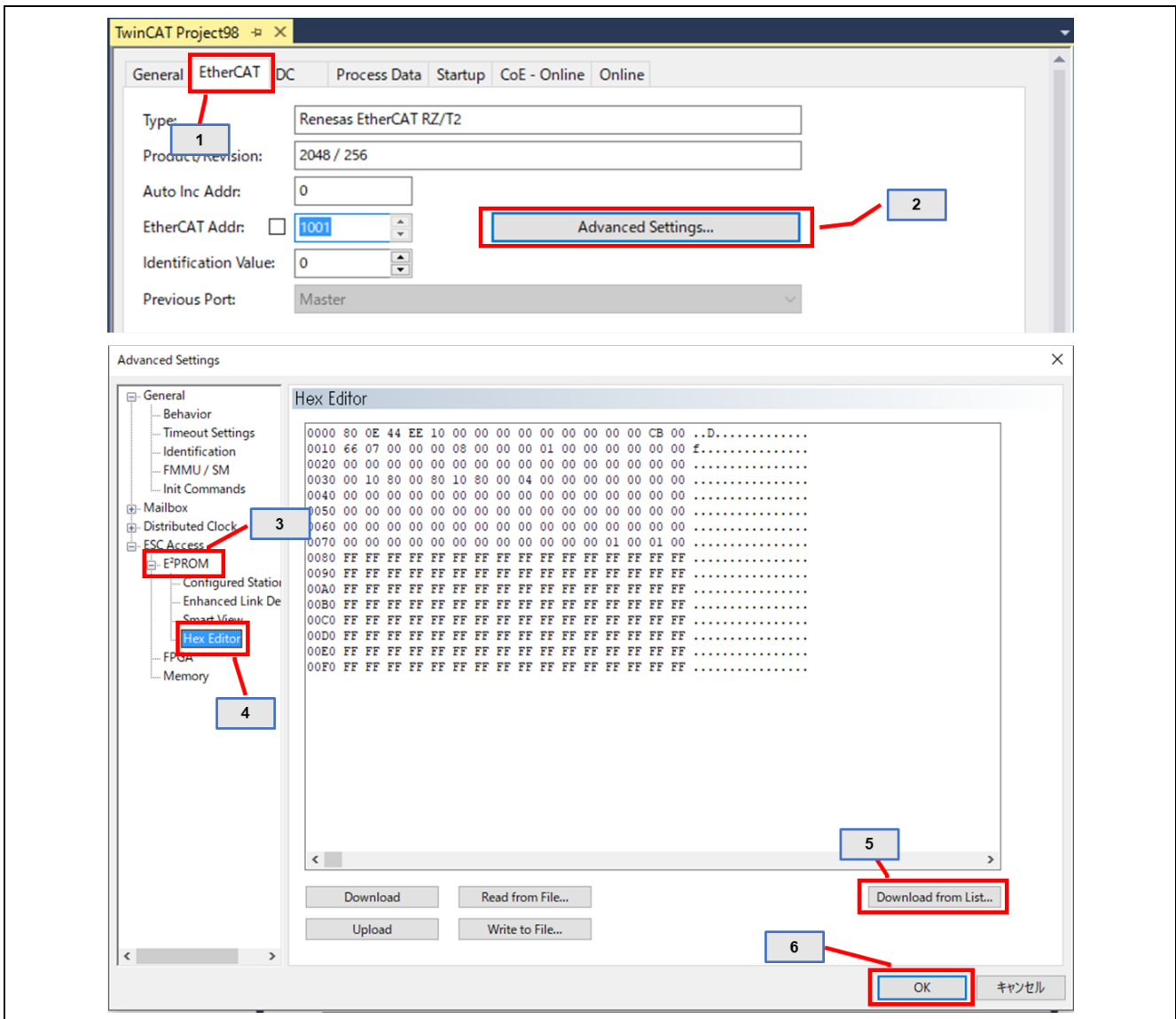
8.2 EEPROM データの更新

他のアプリケーションのデータがすでに EEPROM に書き込まれている場合は、データを置き換えます。EEPROM のデータを置き換える手順を次に示します。

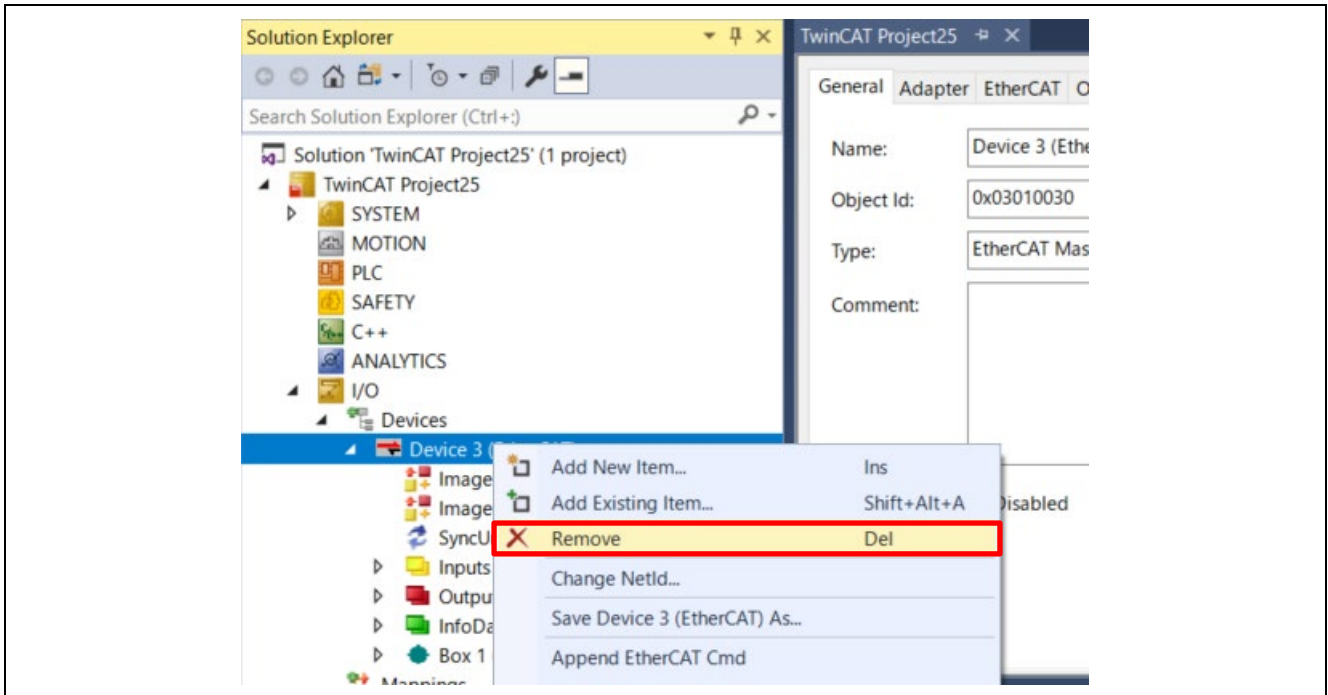
1. [Box 1]をダブルクリックすると、ウィンドウの右側にパネルが表示されます。
2. [EtherCAT]タブを選択します。
3. [Advanced Setting]ボタンをクリックします。
4. [ESC Access]→[EEPROM]→[Hex Editor]を選択します。
5. [[Download from List]を選択
6. “Renesas EtherCAT RA8 2port”を選択



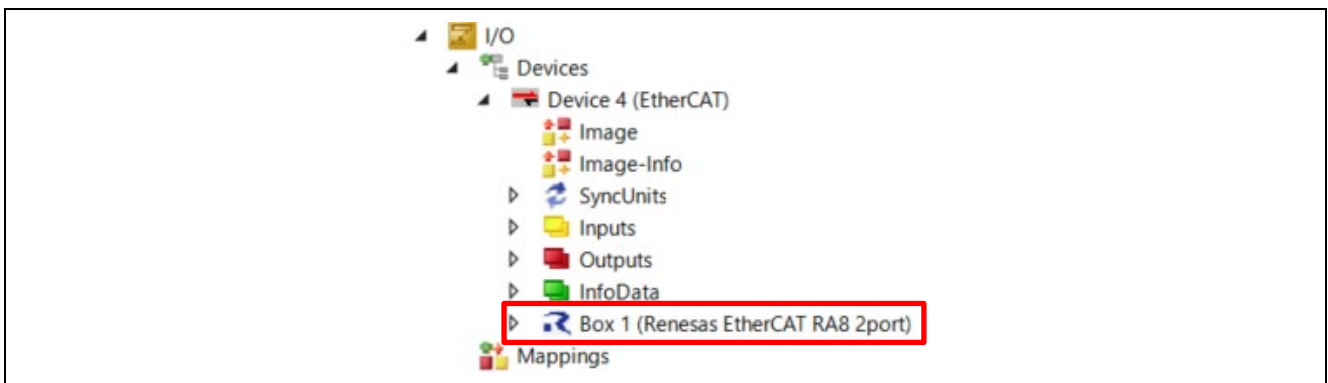
7. OK と [Download] をクリックします。



8. ESI ファイル設定を適用します。
デバイスを選択して削除します。



9. デバイスを再度スキャンします。
ターゲットの ESI ファイルが表示されたら、正しい動作です。



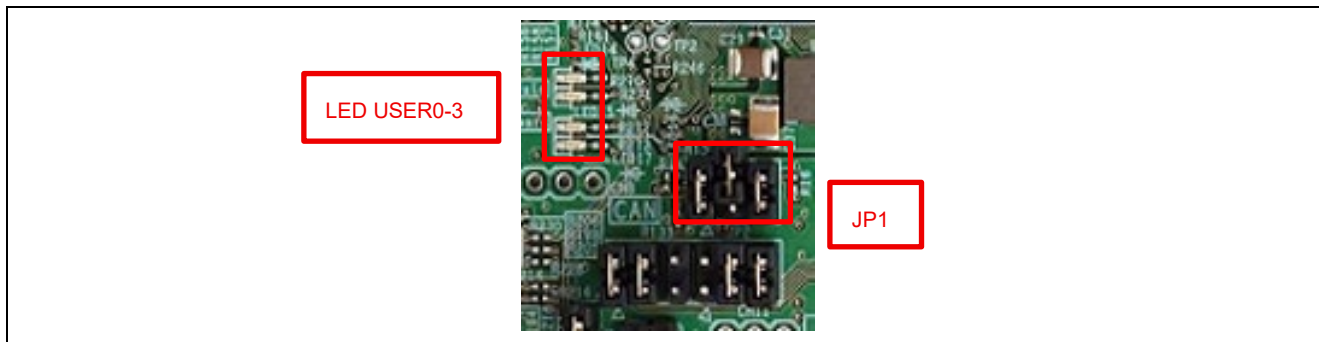
Option A - ESI XML から ESI バイナリファイルを作成し、ダウンロードします。

1. SSC Tool → [Tool] → [EEPROM Programmer].
2. [FILE] → [OPEN] → ESI ファイルを参照して選択します。
3. [FILE] → [Save AS] → タイプをバイナリとして選択。
4. 指定されたフォルダーにバイナリファイルが生成されます。
5. [Read from File] ESI バイナリファイルを選択します。 → [Download].
6. [Upload]オプションを使用して書き込み状態を確認します。

データ差し替え後、RA8T2 を再起動(電源をオフにしてからオン、またはリセット)し、新しいデータが適用されるようにします。[Restart TwinCAT System]を実行します。

8.3 I/O 機能の評価

I/O 機能は、LED_USER0-3 とジャンパ JP1 で確認できます。



1. I/O 出力を確認する場合は、TwinCAT3 で[Output Counte]→[Online]→[Write]を選択、値を入力します。

LED_USER0-3 は 4 ビットの設定値に従って点灯します。

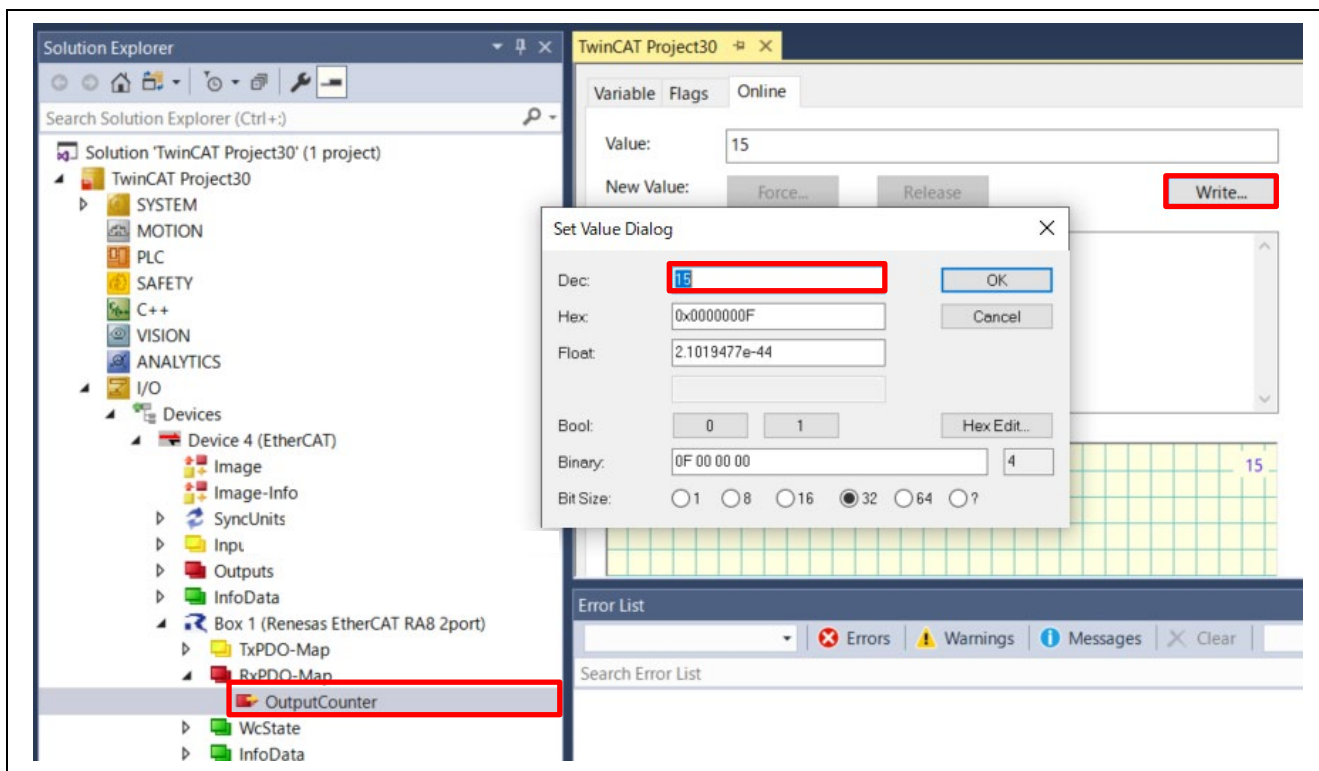


図 2. I/O Output setting

2. I/O 入力を確認する場合は、TwinCAT3 で[Input Counter]→[Online]を選択すると、入力値にジャンパの3ビット入力値が表示されます。

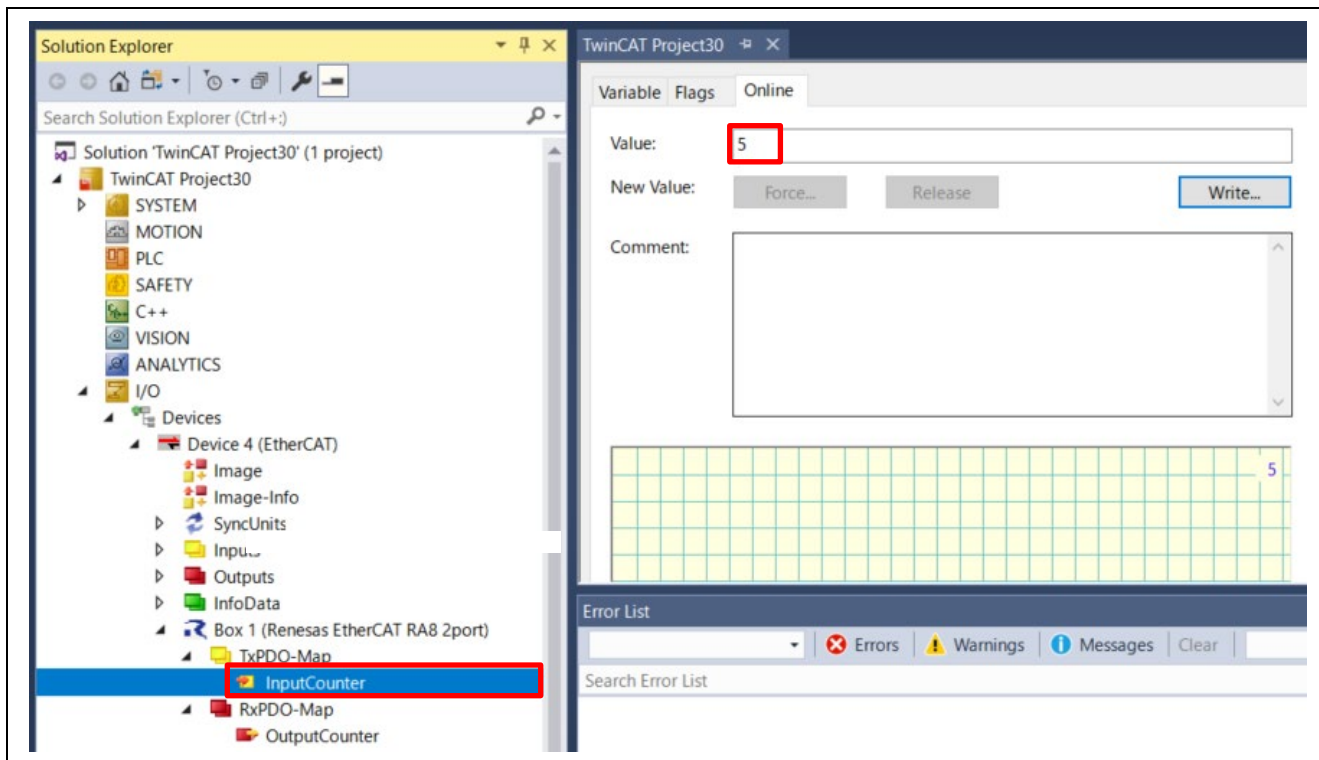
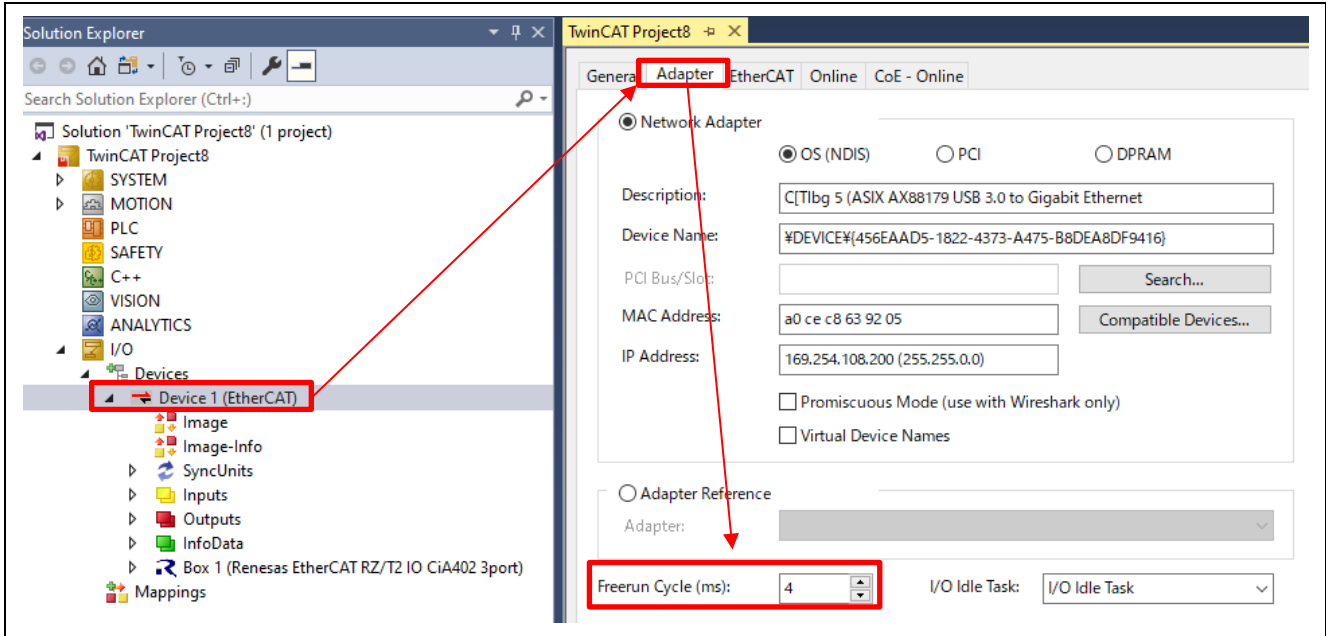


図 3. I/O Input setting

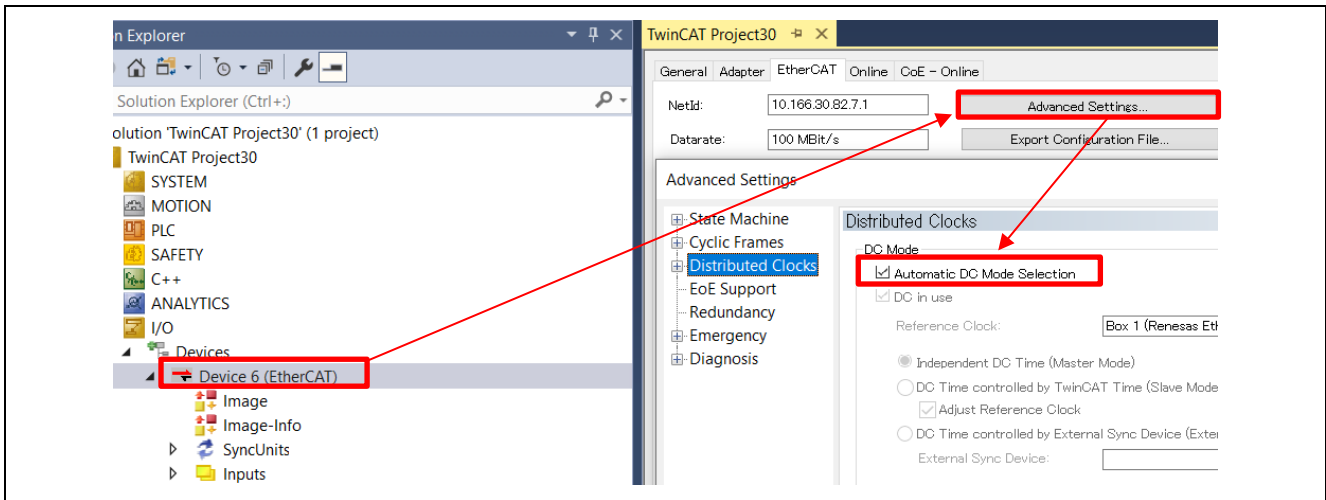
8.4 DC 同期

SyncManager/Sync0 & SyncManager/Sync0/Sync1 同期、AL_EVENT_ENABLED と DC_SUPPORTED の両方が有効になります。

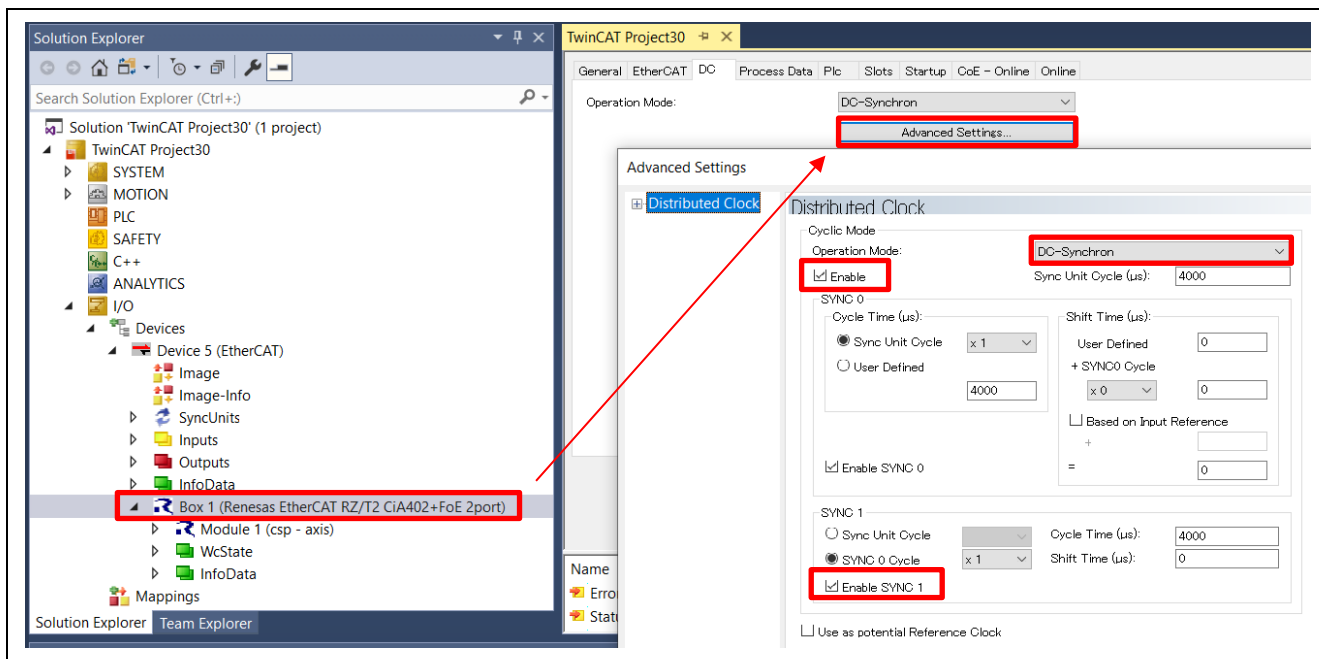
本サンプルプログラムでは、4000μs に設定した Sinc Unit Cycle の動作を確認します。「Adapter」 タブの「Freerun Cycle (ms)」を 4 に変更してください。



1. DC 同期を有効にするためのマスター設定.



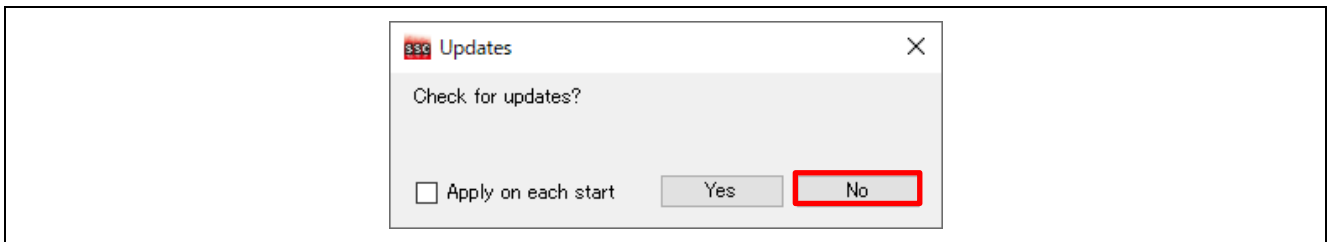
2. DC 同期を有効にするためのサブデバイス設定。



9. Appendix

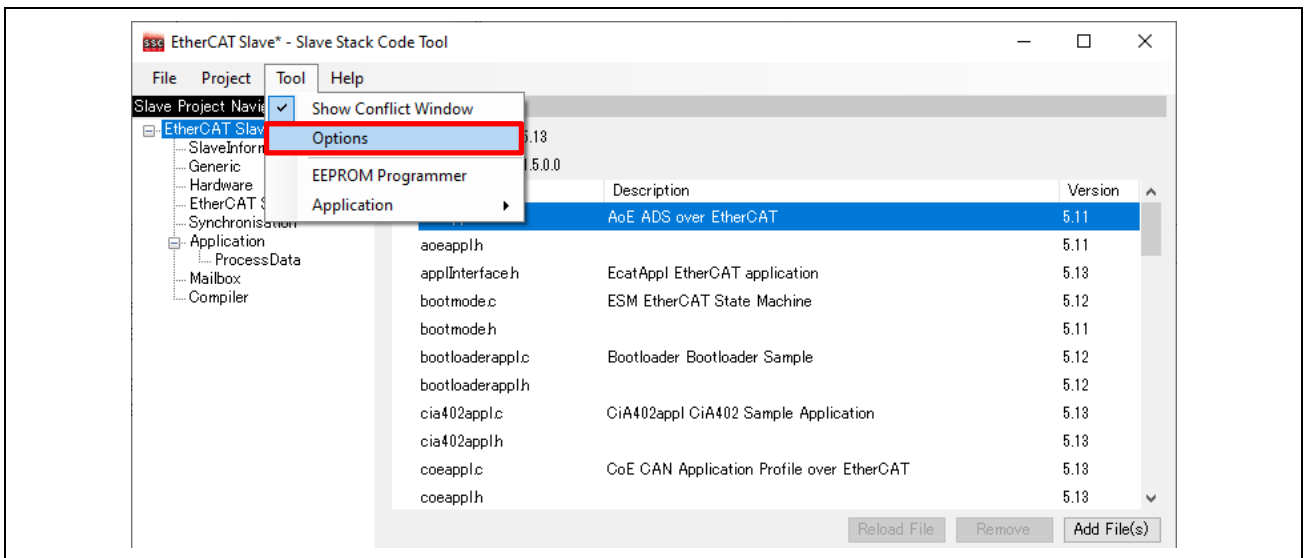
9.1 Appendix A: Point of caution when using SSC Tool

1. SSC ツールを**管理者**として開いてください。
それ以外では、SSC コードの生成が失敗する可能性があります。
2. SSC ツールを初めて開くと、以下のウィンドウが表示されることがあります。
アップデートをチェックしないには、**No** を選択してください。

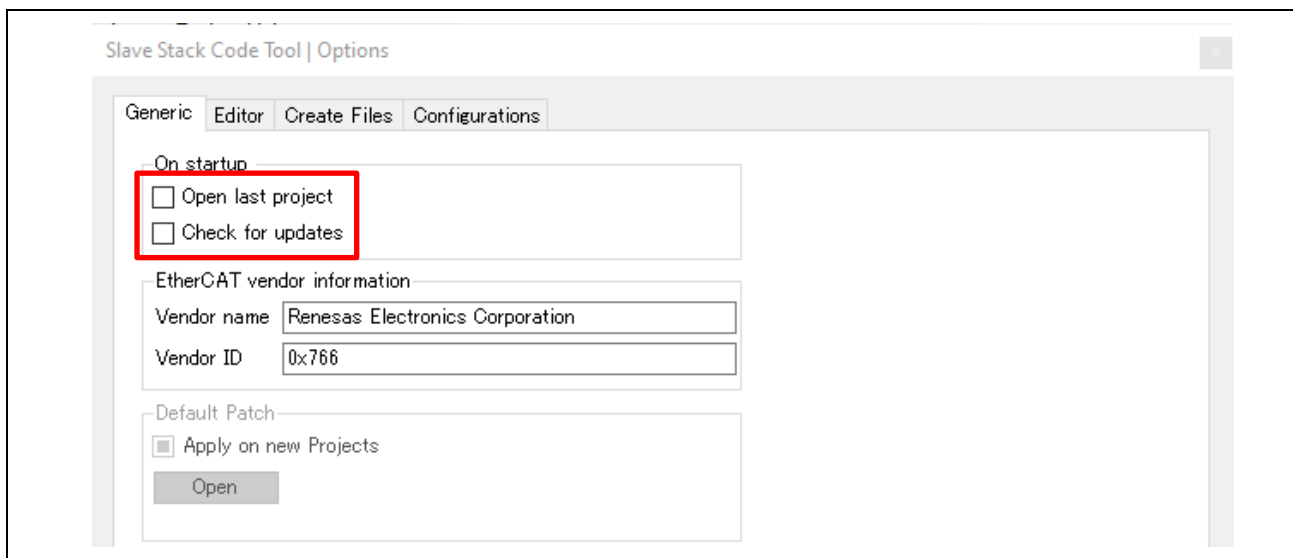


※ Regarding SSC Tool settings

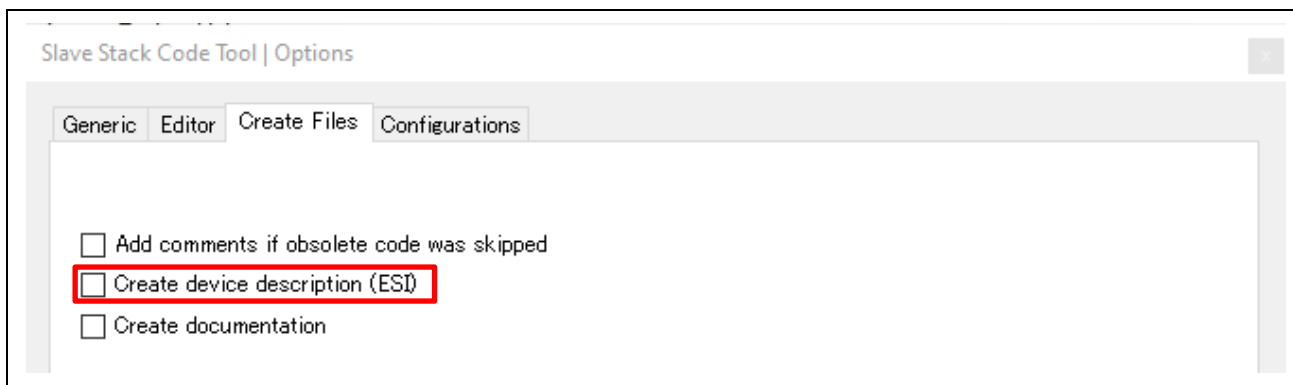
3. [Tool] -> [Options]で SSC ツールの設定を確認してください。



4. [Generic]タブで、[Open last project]と[Check for updates]のチェックを外します。



5. [Create Files]タブで、[Create device description (ESI)]のチェックを外します。



6. [OK]をクリックして、更新した設定を適用します。
これで、このサンプルプログラムの SSC ツールのセットアップは終了です。

9.2 Appendix B : パッチのインストール方法

次の2つの方法があります。

1. Via Git for Windows (64bit)
2. Via MinGW Installation Manager

9.2.1 Using Git for Windows (64bit)

ここでは、Git for Windows 経由でパッチをインストールする方法について説明します。

1. インストーラー (Git-x.xx.x-64-bit.exe など) を Git for Windows の公式 Web サイトからダウンロードします。

[Git for Windows](#)

2. ダウンロードしたインストーラーを実行し、セットアップ手順に従います。特定の要件がない限り、既定の設定を使用します。
3. インストール後、システムの環境変数に patch.exe するパスを追加します。デフォルトのインストールの場合、パスは通常以下ようになります。

“C:¥Users¥<your-username>¥AppData¥Local¥Programs¥Git¥usr¥bin”

変更を適用するには、環境変数を更新した後にコンピュータを再起動します。

コマンドプロンプトを起動し、「where patch」と入力します。

patch.exe へのパスが表示されたら、インストールは成功です。

```
C:¥>where patch
C:¥Users¥          ¥AppData¥Local¥Programs¥Git¥usr¥bin¥patch.exe
```

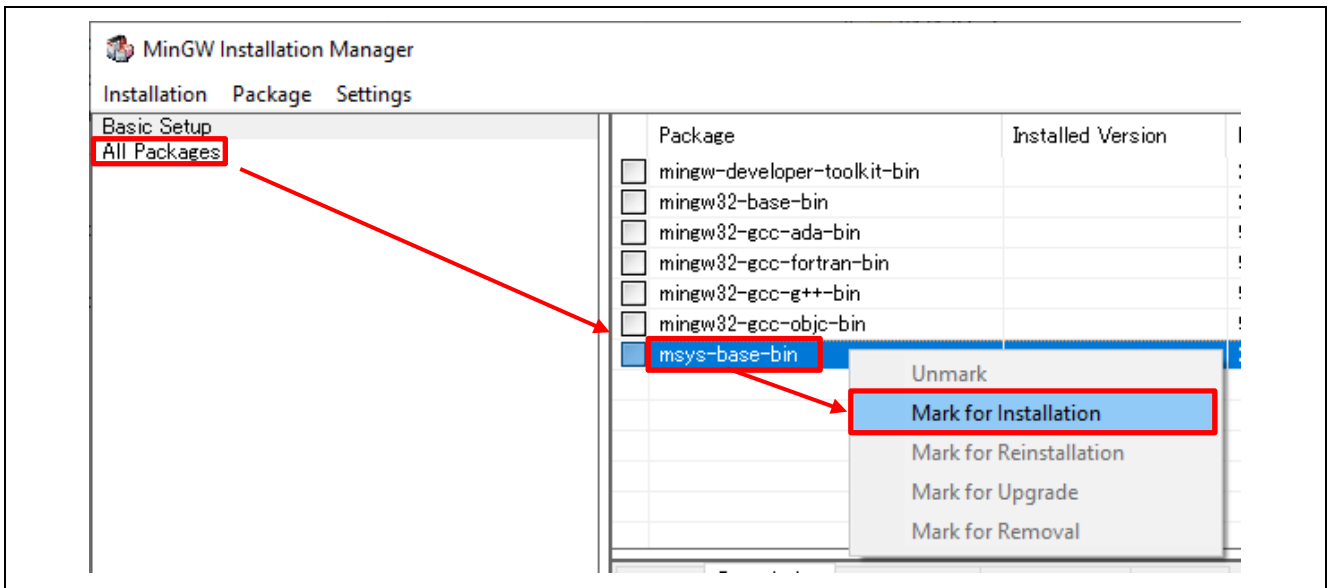
9.2.2 Ushing MinGW Installation Manager

ここでは、MinGW Installation Manager を使用してパッチをインストールする方法について説明します。

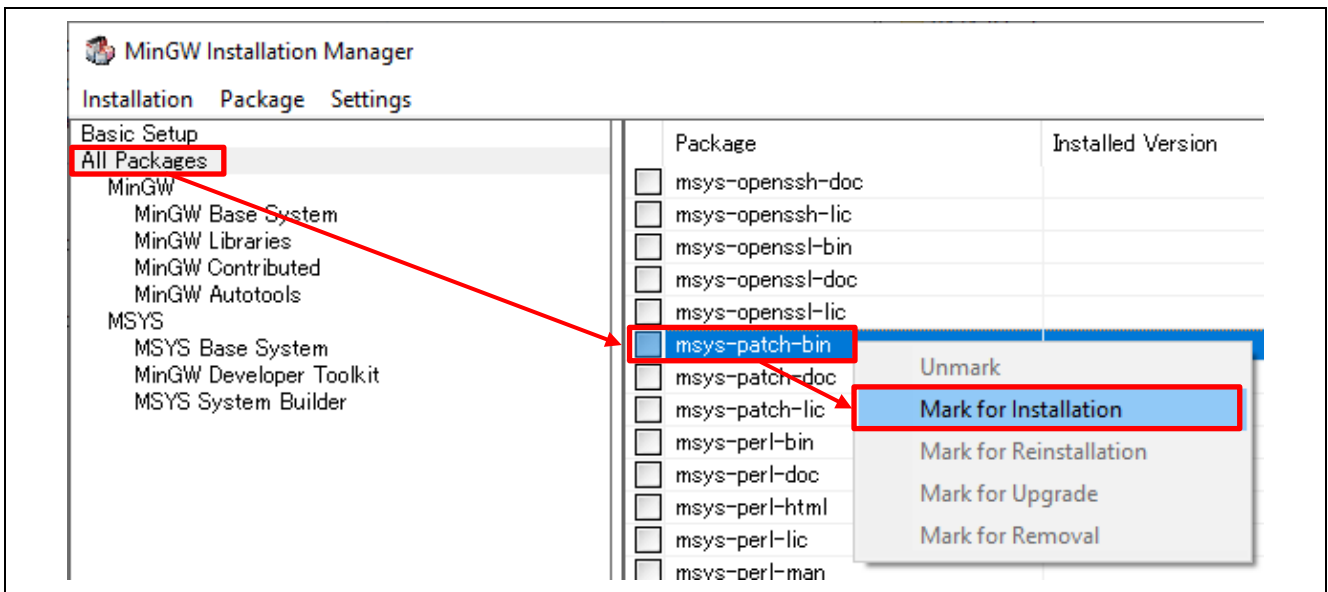
1. 以下の URL から「mingw-get-setup.exe」をダウンロードします。

<https://sourceforge.net/projects/mingw/>

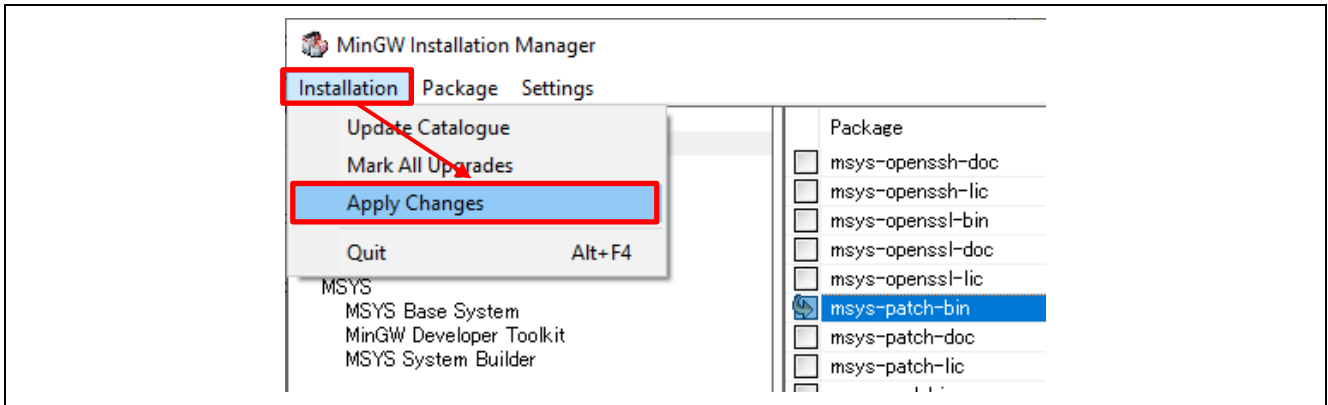
2. 「mingw-get-setup.exe」を実行し、ダイアログに従って「Mingw-installation-manager」をインストールします。
3. 完了後、Mingw-installation-manager ウィンドウが表示されたら、左側のウィンドウで「Basic Setup」を選択し、右側のウィンドウで「msys-base-bin」を右クリックして、「Mark for Installation」を選択します。



4. 左側のウィンドウで「All Packages」を選択し、右側のウィンドウで「msys-patch-bin」を右クリックして、「Mark for Installation」を選択します。

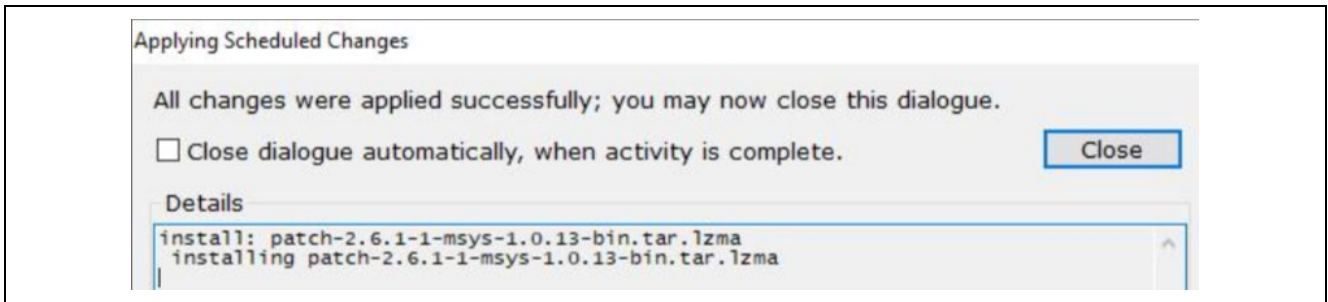


5. メニューバーの「Installation」で「Apply Changes」を選択します。



6. 「Schedule of Pending Actions」ウィンドウが表示されたら、「Apply」ボタンをクリックします。

7. 「All changes were applied successfully」と表示され、インストール patch.exe は成功です

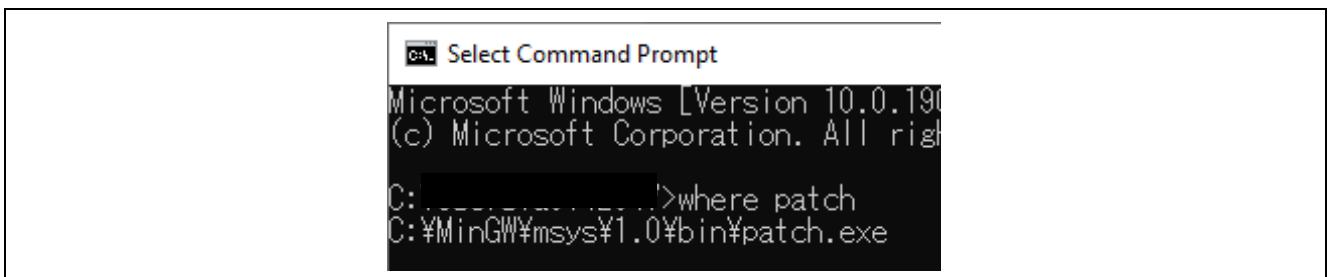


8. インストールされている patch.exe へのパスをシステム環境変数に追加します。
例として、デフォルトパスの場合は次のパスを追加します。

“C:¥MinGW¥msys¥1.0¥bin”

システム環境変数を更新した後、コンピュータを再起動して変更を適用します。

9. コマンドプロンプトを起動し、「where patch」と入力します。
patch.exe へのパスが表示されたら、インストールは成功です。



9.3 Appendix C : EEPROM 容量設定変更

スマート コンフィギュレータを使用して、EEPROM の容量サイズを変更し、コードを生成します。

EtherCAT の仕様では、32Kbit を境界として設定を変更する必要があります。

1. スタック SSC_port 選択します。[Properties] → [EEPROM Size]
2. EEPROM の容量に応じて、「Under 32Kbits」または「Over 32Kbits」を選択します。

The screenshot shows the 'Stacks Configuration' interface. The 'HAL/Common Stacks' section contains several stacks, with 'g_ethercat_ssc_port0 EtherCAT SSC Port (rm_ethercat_ssc_port)' highlighted. A red box surrounds this stack, and a red arrow points to the 'Properties' tab in the 'g_ethercat_ssc_port0 EtherCAT SSC Port (rm_ethercat_ssc_port)' settings panel. In the 'Properties' panel, the 'EEPROM Size' property is highlighted, and a dropdown menu is open, showing three options: 'Under 32Kbits', 'Under 32Kbits', and 'Over 32Kbits'. A red box surrounds the dropdown menu, and a red arrow points to the 'Under 32Kbits' option.

3. 「Generate Project Content」でコードを生成します。

10. 制限事項

1. 本サンプルプログラムでは、「9.2 Appendix B」に記載のパッチは使用しないため、パッチのインストールや設定は不要です。

改訂記録

| Rev. | 発行日 | 改訂内容 | |
|------|----------|------|------|
| | | ページ | ポイント |
| 1.00 | 2025.9.8 | - | 初版発行 |
| | | | |

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. 静電気対策

CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。

2. 電源投入時の処置

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. 電源オフ時における入力信号

当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れしないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。

4. 未使用端子の処理

未使用端子は、「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。

5. クロックについて

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。

6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、 V_{IL} (Max.) から V_{IH} (Min.) までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、 V_{IL} (Max.) から V_{IH} (Min.) までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

7. リザーブアドレス（予約領域）のアクセス禁止

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、将来の拡張機能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。

8. 製品間の相違について

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えば、フラッシュメモリ、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。全著作権所有。

○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、Institute of Electrical and Electronics Engineers Inc.の登録商標です

○EtherCAT[®]およびTwinCAT[®]は、ドイツの Beckhoff Automation GmbH によってライセンスされた登録商標および特許技術です。

○J-Link は、SEGGER Microcontroller GmbH の商標です。

○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まれます。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)

本社所在地

〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。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。