

# NX シリーズ 開発チュートリアル

1 版 2025 年 05 月 16 日

|   |           |
|---|-----------|
| <b>1. 概要</b> .....  | <b>2</b>  |
| 1.1 概要 .....  | 2         |
| 1.2 使用環境 .....  | 2         |
| 1.3 CPU ボード スイッチ設定 .....  | 4         |
| 1.4 各 CPU ボードの資料に関して .....  | 5         |
| <b>2. 機能</b> .....  | <b>6</b>  |
| 2.1 サンプルプログラムの使用機能 .....  | 6         |
| <b>3. MCUXpresso IDE を用いた動作方法</b> .....                             | <b>7</b>  |
| 3.1 サンプルプログラムの動作方法 .....  | 7         |
| 3.1.1 SDK のインストール方法 .....   | 7         |
| 3.1.2 インポート方法 .....   | 10        |
| 3.1.3 ビルド構成の変更方法 .....  | 13        |
| 3.1.4 ビルド方法 .....   | 14        |
| 3.1.5 内蔵 RAM を用いたデバッグ、ダウンロード方法 .....                                | 15        |
| 3.1.6 外部 ROM を用いたデバッグ、ダウンロード方法 .....                                | 20        |
| 3.2 サンプルプログラムに機能を追加修正する際の注意事項 .....                                 | 25        |
| <b>4. MCUXpresso IDE を用いた新規プロジェクト作成方法</b> .....                     | <b>26</b> |
| 4.1 内蔵 RAM 上で動作するプログラムの作成例 .....                                    | 26        |
| 4.2 外部 ROM 上で動作するプログラムの作成例 .....                                    | 36        |
| 4.2.1 NX-RT1021、NX-RT1062 の変更内容 .....                               | 36        |
| 4.2.2 NX-N947 の変更内容 .....   | 36        |
| 4.3 SDRAM を使用するプログラムの作成例 .....                                      | 38        |
| 4.3.1 SEMC の設定 .....  | 38        |
| 4.3.2 ポートの設定 .....  | 42        |
| 4.3.3 DCD の設定 .....   | 43        |
| <b>5. MCUXpresso Secure Provisioning を使用した外部 FlashROM への書き込み</b> .. | <b>46</b> |
| 5.1 USB インタフェースを使用した書き込み方法 .....                                    | 46        |

## 1. 概要

### 1.1 概要

本アプリケーションノートは、弊社製 NX シリーズ CPU ボードを用いて、MCUXpresso SDK を使用したプログラムを作成・動作確認するまでのチュートリアル資料です。

各機能に関する詳細は、サンプルプログラムのソースコードやサンプルプログラム解説、ハードウェアマニュアル、回路図、および、CPU のリファレンスマニュアル、NXP 社製開発ツールをインストールしたフォルダ内にあるユーザガイドなどの各種マニュアル (pdf ファイル) をご覧ください。

### 1.2 使用環境

本アプリケーションノートの解説で用いる開発環境を以下に示します。

<NX-RT1021>

| ソフトウェア         | バージョン             | 備考                          |
|----------------|-------------------|-----------------------------|
| MCUXpresso IDE | V24.12[Build 148] | –                           |
| MCUXpresso SDK | V24.12.00         | SDK_2.x_MIMXRT1021xxxxx を使用 |

| デバッガ     | ハードウェアバージョン | 備考 |
|----------|-------------|----|
| MCU-Link | V3.146      | –  |

| Flash 書き込みツール                  | ハードウェアバージョン    | 備考 |
|--------------------------------|----------------|----|
| MCUXpresso Secure Provisioning | V10.0[b241106] | –  |

<NX-RT1062 >

| ソフトウェア         | バージョン             | 備考                          |
|----------------|-------------------|-----------------------------|
| MCUXpresso IDE | V24.12[Build 148] | –                           |
| MCUXpresso SDK | V24.12.00         | SDK_2.x_MIMXRT1062xxxxB を使用 |

| デバッガ     | ハードウェアバージョン | 備考 |
|----------|-------------|----|
| MCU-Link | V3.146      | –  |

| Flash 書き込みツール                  | ハードウェアバージョン    | 備考 |
|--------------------------------|----------------|----|
| MCUXpresso Secure Provisioning | V10.0[b241106] | –  |

&lt;NX-N947&gt;

| ソフトウェア         | バージョン             | 備考                  |
|----------------|-------------------|---------------------|
| MCUXpresso IDE | V24.12[Build 148] | -                   |
| MCUXpresso SDK | V24.12.00         | SDK_2.x_MCXN947 を使用 |

| デバッガ     | ハードウェアバージョン | 備考 |
|----------|-------------|----|
| MCU-Link | V3.146      | -  |

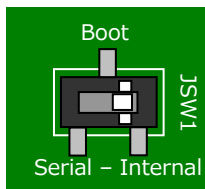
| Flash 書き込みツール                  | ハードウェアバージョン    | 備考 |
|--------------------------------|----------------|----|
| MCUXpresso Secure Provisioning | V10.0[b241106] | -  |

### 1.3 CPU ボード スイッチ設定

CPU ボードを動作させる際は、動作モードに応じてボード上のディップスイッチを設定する必要があります。

#### (1) NX-RT1021

出荷時設定：「Internal Boot モード」



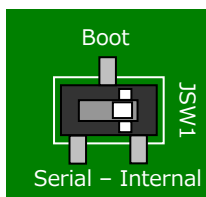
BootMode

Serial : Serial Downloader モード

Internal : Internal Boot モード

#### (2) NX-RT1062

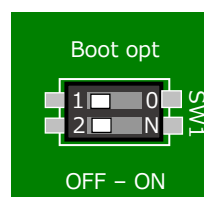
出荷時設定：「Internal Boot モード」および「QSPI Flash Boot,Encryped を使用しない」



BootMode

Serial : Serial Downloader モード

Internal : Internal Boot モード



BootOption

SW1.1-1.2

[OFF,OFF]:QSPI Flash Boot,Encryped を使用しない場合

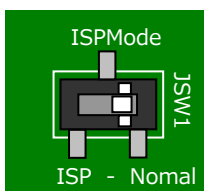
[OFF,ON] :QSPI Flash Boot,Encryped を使用する場合

[ON,OFF] :uSDHC1 Boot を使用する場合

[ON,ON] :設定禁止

#### (3) NX-N947

出荷時設定：「Nomal(Master Boot モード)」



ISPMode

ISP : ISP モード

Nomal : Master Boot モード

## 1.4 各 CPU ボードの資料に関して

資料・サンプルプログラムは、弊社 Web サイトのボード紹介ページで公開されています。

|                 |   |
|-----------------|---|
| NXP CPU ボードシリーズ | <a href="https://www.apnet.co.jp/product/nx/index.html">https://www.apnet.co.jp/product/nx/index.html</a>         |
| NX-RT1021 製品ページ | <a href="https://www.apnet.co.jp/product/nx/nx-rt1021.html">https://www.apnet.co.jp/product/nx/nx-rt1021.html</a> |
| NX-RT1062 製品ページ | <a href="https://www.apnet.co.jp/product/nx/nx-rt1062.html">https://www.apnet.co.jp/product/nx/nx-rt1062.html</a> |
| NX-N947 製品ページ   | <a href="https://www.apnet.co.jp/product/nx/nx-n947.html">https://www.apnet.co.jp/product/nx/nx-n947.html</a>     |

## 2. 機能

### 2.1 サンプルプログラムの使用機能

以下のサンプルプログラムを用いて、各機能の確認をすることができます。

(2025年5月現在の公開データの状況です)

#### <NX-RT1021>

| ドキュメント番号 | サンプルプログラム      | 機能   |
|----------|----------------|--|
| AN2302   | 標準サンプルプログラム    | <ul style="list-style-type: none"> <li>・ CAN 通信</li> <li>・ UART 通信</li> <li>・ USB ファンクション 仮想 COM 通信</li> </ul> |
| AN2303   | Pmod サンプルプログラム | <ul style="list-style-type: none"> <li>・ Pmod 動作<br/>(GPIO/UART/SPI/I2C)</li> </ul>                            |

#### <NX-RT1062>

| ドキュメント番号 | サンプルプログラム         | 機能   |
|----------|-------------------|--|
| AN2302   | 標準サンプルプログラム       | <ul style="list-style-type: none"> <li>・ CAN 通信</li> <li>・ ネットワーク通信</li> <li>・ UART 通信 SDRAM 初期化</li> <li>・ SD カード読み書き</li> <li>・ USB ホスト メモリ読み書き</li> <li>・ USB ファンクション 仮想 COM 通信</li> <li>・ audio 入出力</li> </ul> |
| AN2303   | Pmod サンプルプログラム    | <ul style="list-style-type: none"> <li>・ Pmod 動作<br/>(GPIO/UART/SPI/I2C)</li> </ul>  |
| AN2304   | LCD-KIT サンプルプログラム | <ul style="list-style-type: none"> <li>・ LCD 出力</li> <li>・ タッチネル操作</li> </ul>  |

#### <NX-N947>

| ドキュメント番号 | サンプルプログラム    | 機能   |
|----------|--------------|--|
| AN2305   | 標準サンプルプログラム  | <ul style="list-style-type: none"> <li>・ CAN 通信</li> <li>・ ネットワーク通信</li> <li>・ UART 通信 PSRAM 初期化</li> <li>・ USB ホスト メモリ読み書き</li> <li>・ USB ファンクション 仮想 COM 通信</li> <li>・ audio 入出力</li> </ul> |
| AN2306   | AI サンプルプログラム | <ul style="list-style-type: none"> <li>・ 画像分類</li> </ul>   |

### 3. MCUXpresso IDE を用いた動作方法

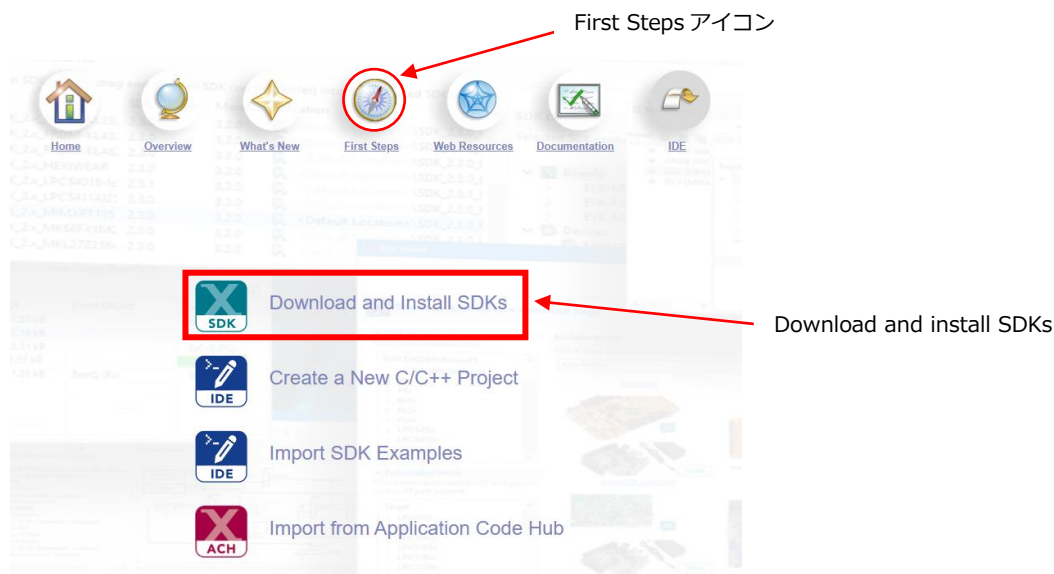
本章では、MCUXpresso IDE を使用して CPU ボードを動作させる方法を説明します。  
MCUXpresso IDE はインストール済みであるものとしますので、事前にインストールを行ってください。  
なお、本書では、インストール後、起動をしていない状態を前提として説明します。

#### 3.1 サンプルプログラムの動作方法

本節では、弊社で作成した NX-RT1062 サンプルプログラムを例に動作方法の説明をいたします。  
弊社から取得したサンプルプログラム(zip)を解凍し、ワークスペースに移動(コピー)してください。  
本書では、「nx\_rt1062\_usb\_pcdc\_sample」(USB ファンクション 仮想 COM 通信)を用いて説明を進めます。  
手順としては、1.SDK インストール、2.サンプルプログラムのインポート、3.ビルド、4.CPU ボードへの書き込みとなります。

##### 3.1.1 SDK のインストール方法

- ① MCUXpresso を起動します。  
起動に成功した場合、Welcome ページが表示されます。(初回の起動、または、新規ワークスペースディレクトリを設定した後)  
Welcome ページにある「Download and install SDKs」を選択し、SDK をインストールしてください。  
なお、「First Steps」アイコンを選択すると SDK のインストール方法が表示されます。



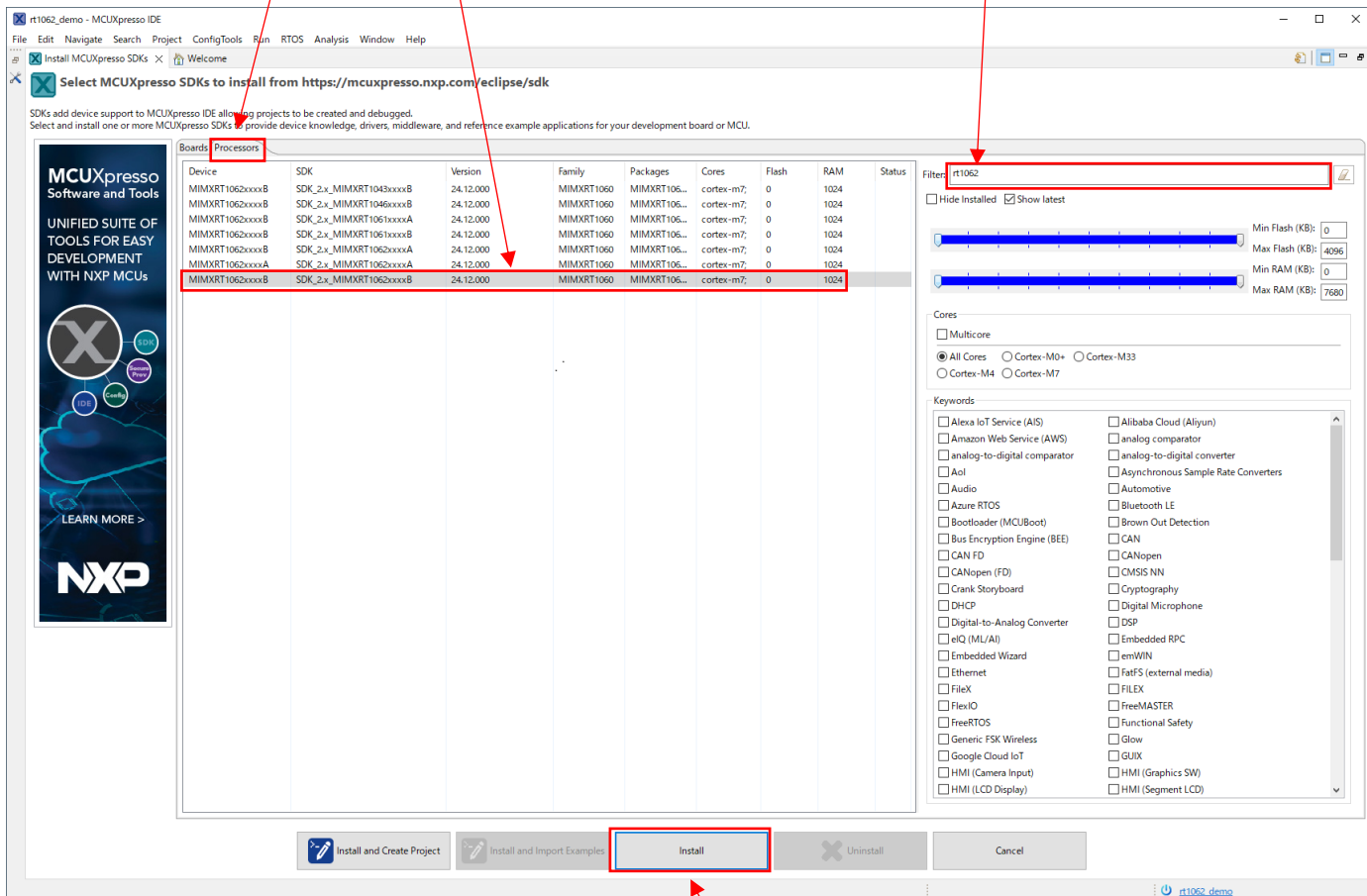
インストールする SDK は「1.2 使用環境について」を参照してください。

- ② Processors タブの中から、" SDK\_2.x\_MIMXRT1062xxxxB" を選択し、SDK をインストールしてください。  
 (CPU ボード毎の SDK は、1.2 項を参照ください)

(1) Processors タブを選択

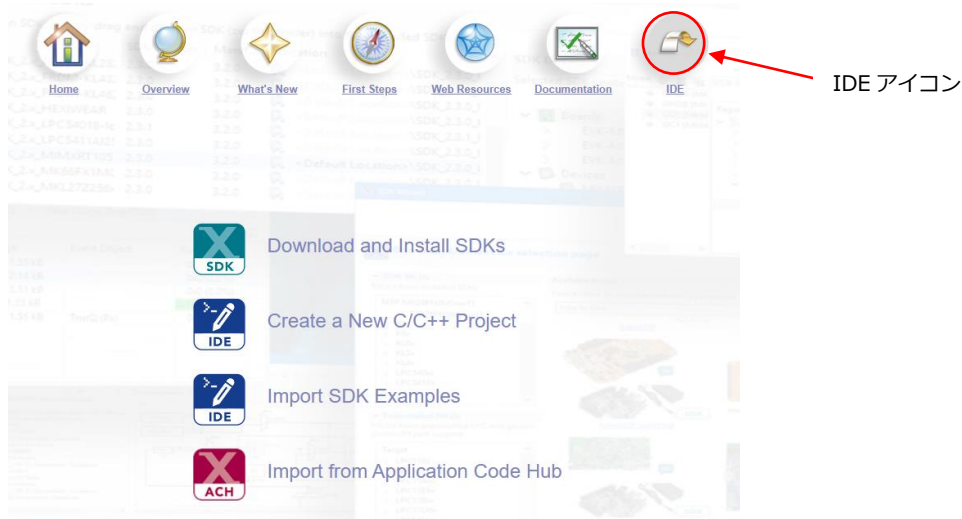
(3) SDK\_2.x\_MIMXRT1062xxxxB を選択

(2) フィルタに"rt1021","rt1062","n947"を  
 入れて選択肢を絞ります

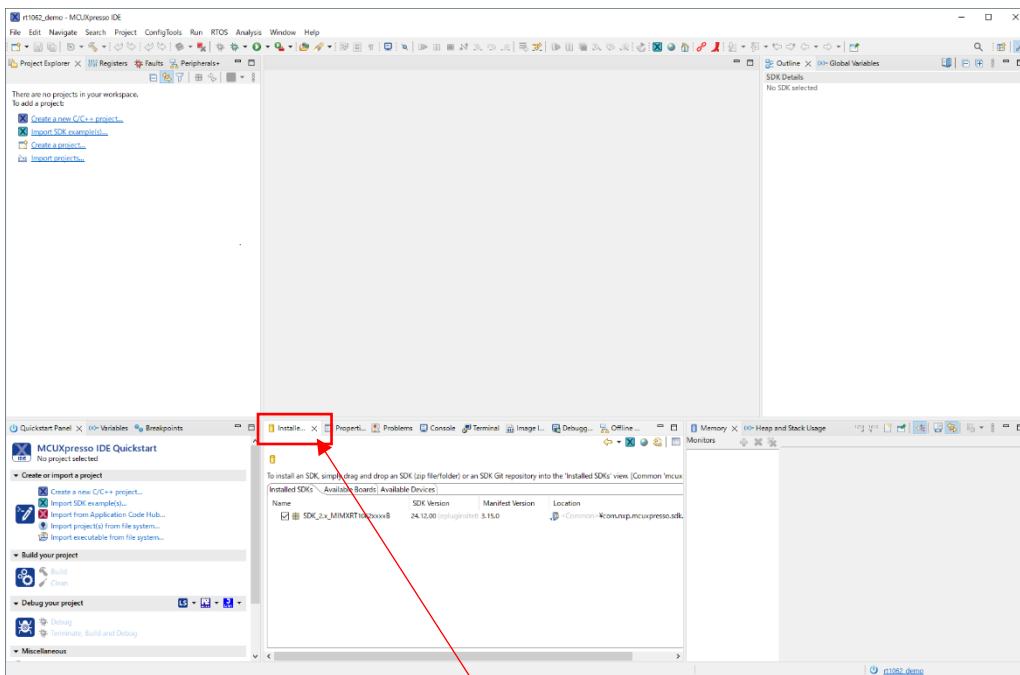


(4) 「Install」 をクリックし、SDK のインストールを開始します

③ SDK をインストール完了した後、**IDE アイコン**をクリックし IDE 画面に移動してください。



IDE に移行した直後の画面

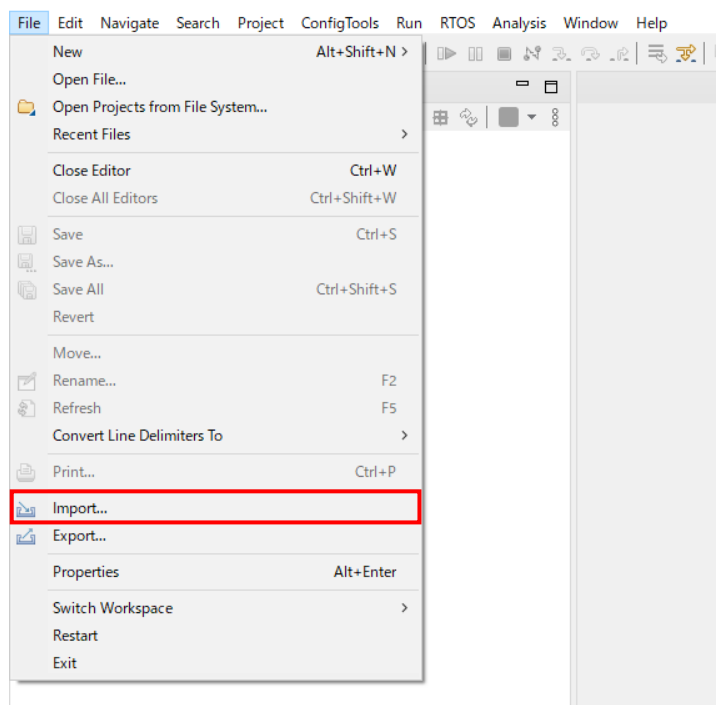


正常に SDK インストールが完了した場合、「Installed SDKs」のタブでインストールした SDK の確認が行えます。

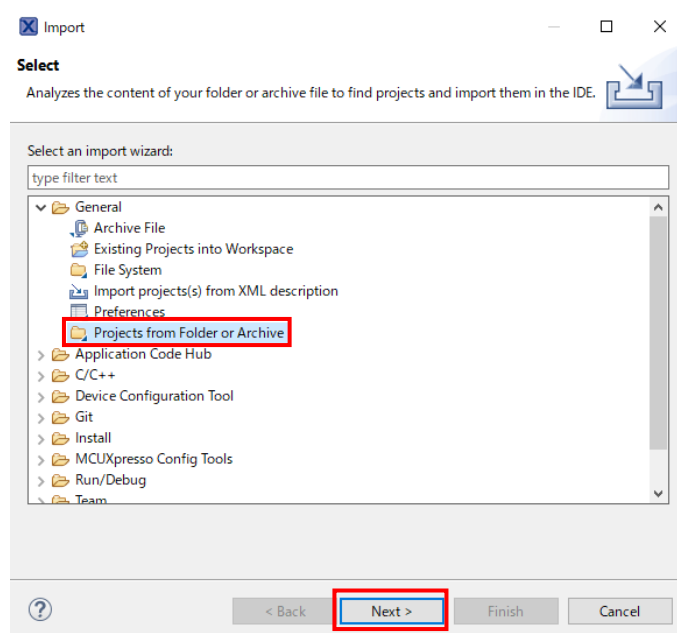
| Installed SDKs  |                        | Available Boards | Available Devices               |
|---|------------------------|------------------|---------------------------------|
| Name  | SDK Version            | Manifest Version | Location                        |
| <input checked="" type="checkbox"/> SDK_2.x_MIMXRT1062xxxxB | 24.12.00 (pluginsite@) | 3.15.0           | <Common>%com.nxp.mcuxpresso.sdk |

## 3.1.2 インポート方法

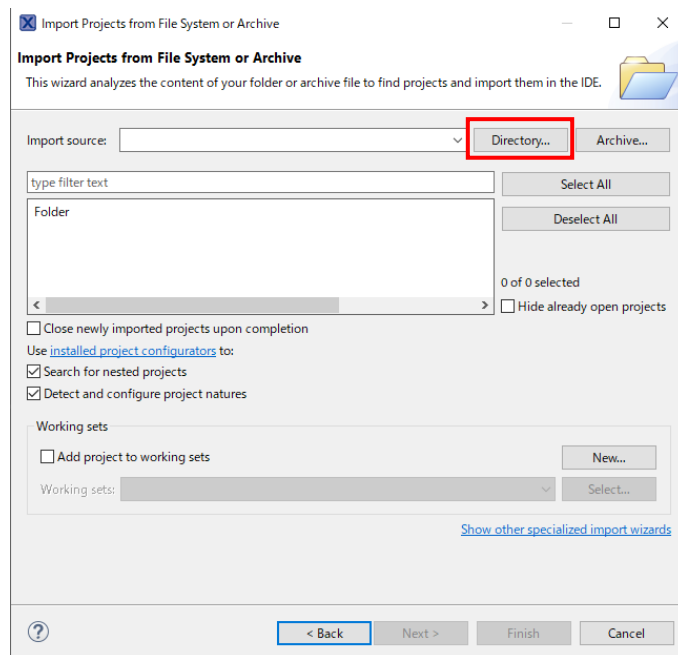
- ① ツールバーの [file] → [import] を選択します。



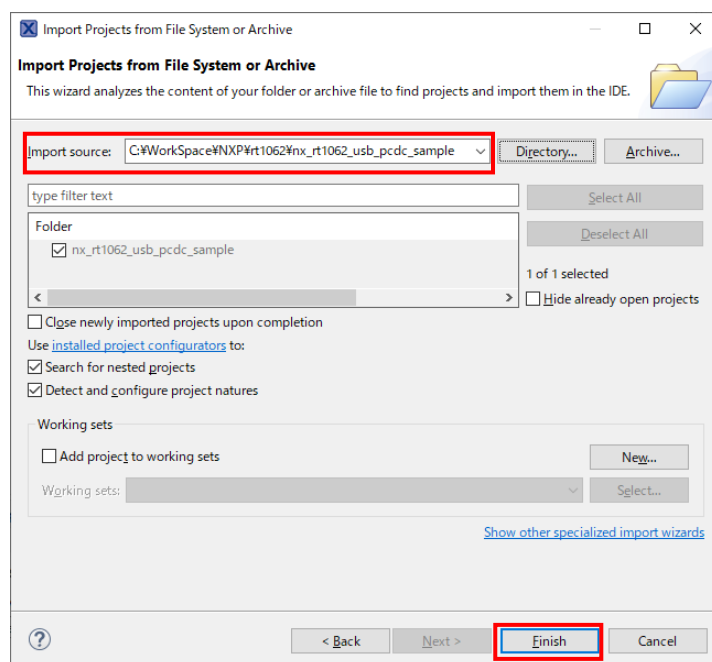
- ② [General] → [Projects from Folder or Archive] を選択し、[Next >]をクリックします。



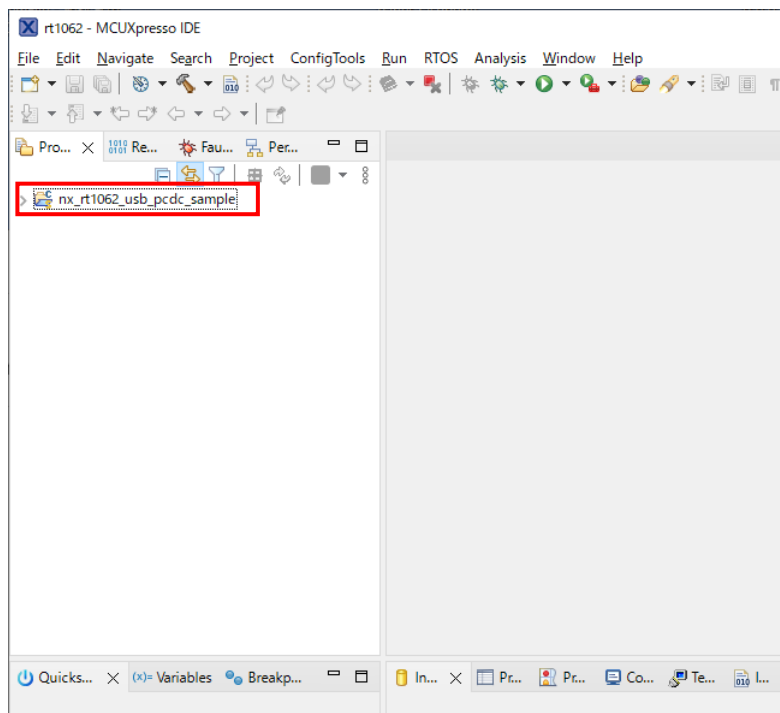
- ③ [Directory...] をクリックし、サンプルプログラムのフォルダーを選択します。



- ④ [Import source]にて使用する Sample が確認できましたら、[Finish] をクリックします。



- ⑤ [Project Explorer] にサンプルプロジェクトが追加されていることを確認します。



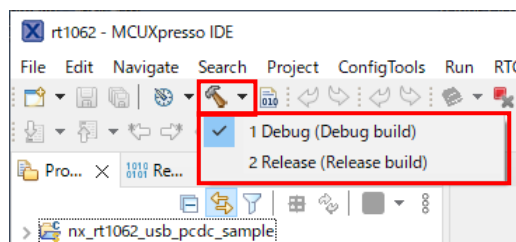
以上でプロジェクトのインポートは完了です。

### 3.1.3 ビルド構成の変更方法

弊社から提供しているサンプルプログラムには、Debug と Release、2種類のビルド構成があります。

Debug は、RAM 上にプログラムをダウンロードする構成となっており、Release は、FlashROM へプログラムをダウンロードする構成となっております。

ビルド構成を変更は、ツールバーにあるビルドアイコンの [▼] をクリックすることで表示されるプルダウンメニューから使用したいビルド構成をクリックすることで行えます。



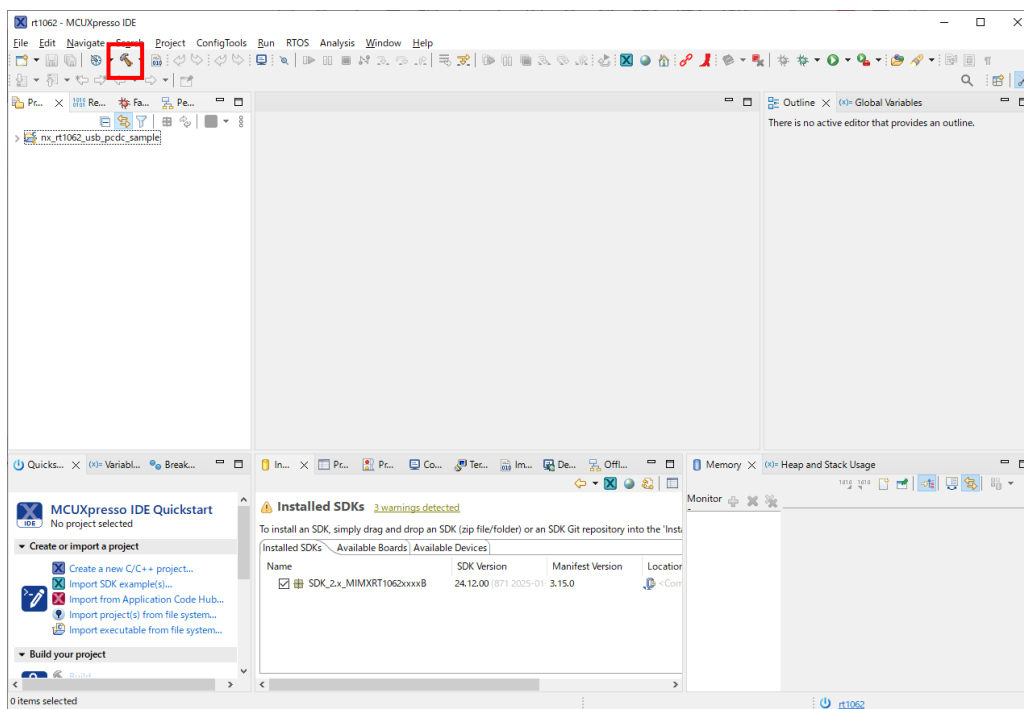
本サンプルプログラムでは、「Release」は、FlashROM へ書き込まれる構成となり、XIP (Execute-in-Place) モードにて動作します。

そのため、MCUXpresso IDE からのダウンロード(FlashROM への書き込み)が行え、デバッグも可能です。

また、デバッグが終了後、ボードのみで起動させることもできます。

### 3.1.4 ビルド方法

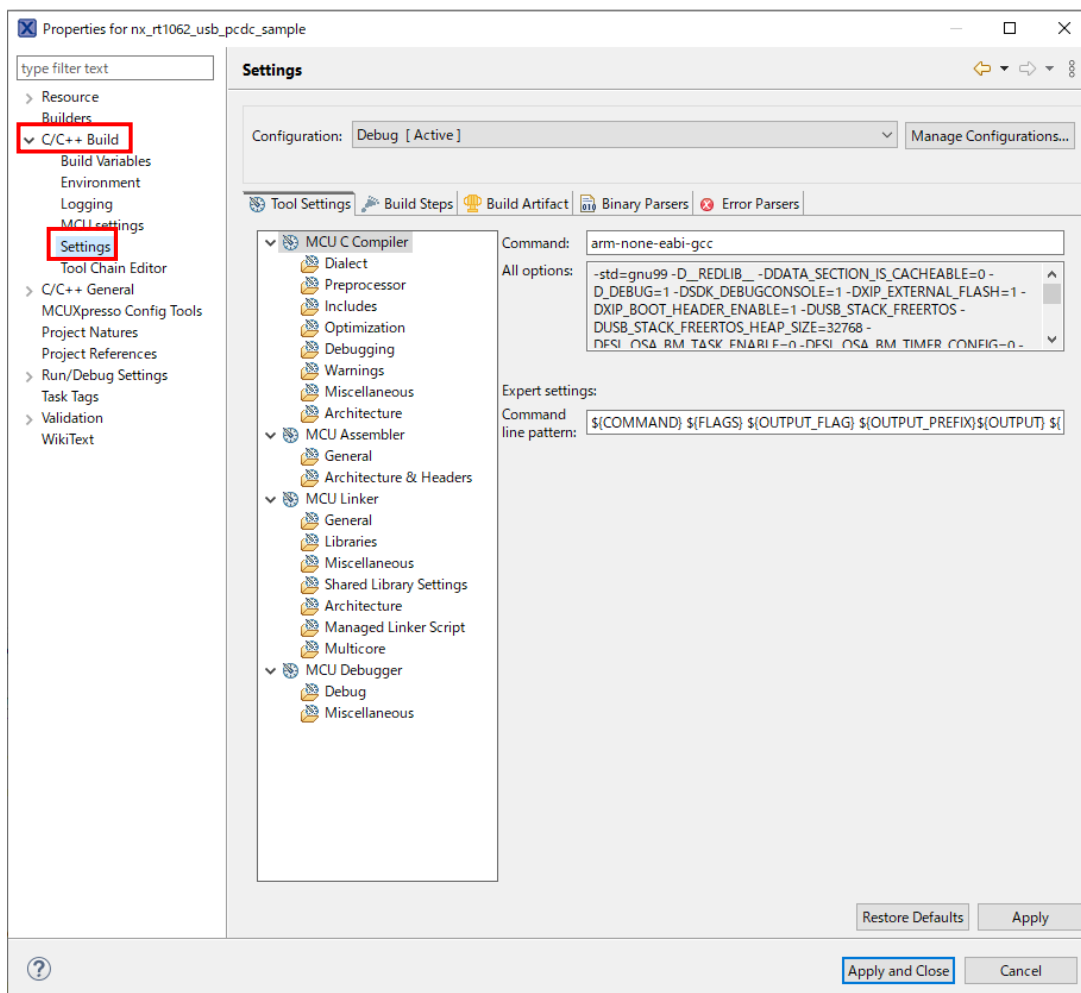
- ① ビルド構成を選択します。  
ビルドを行うビルド構成 (Debug、もしくは、Release) を選択します。
- ② ツールバーのビルドアイコンをクリックします。  
ビルドが成功すると、¥Debug ワークフォルダにオブジェクトファイルが生成されます。



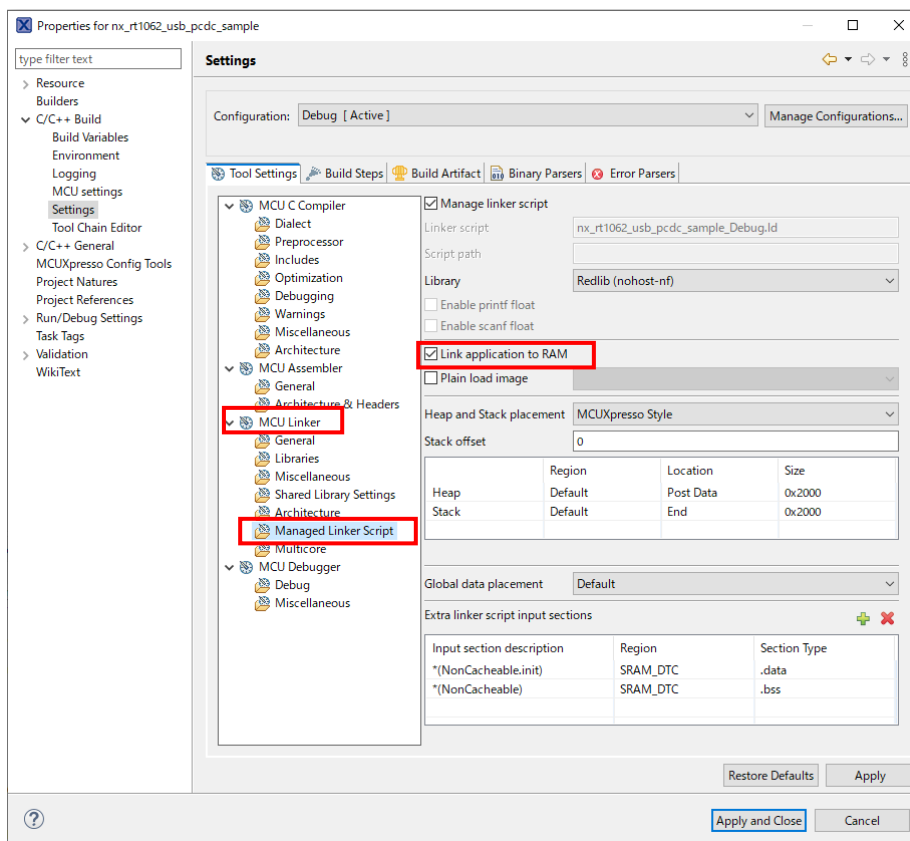
MCUXpresso IDE の詳細な使用方法に関しては、MCUXpresso IDE のマニュアル類を参照してください。

## 3.1.5 内蔵 RAM を用いたデバッグ、ダウンロード方法

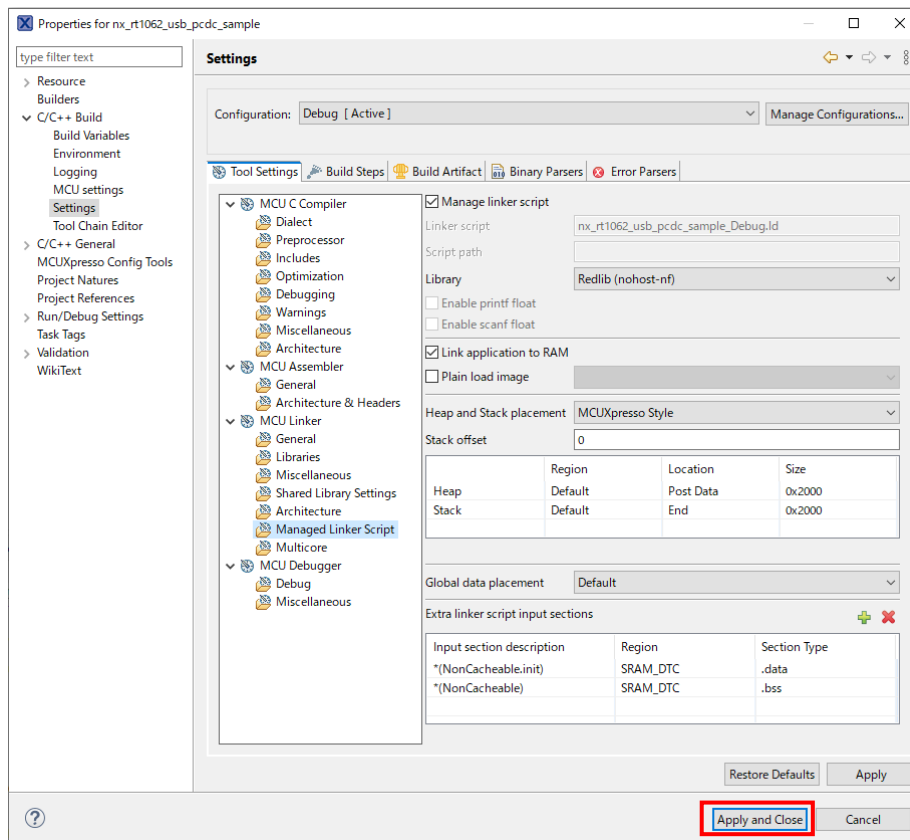
- ① ビルド構成に [Debug] を選択します。
- ② Project Explorer にあるプロジェクトを右クリックし、[Properties]を開き、[C/C++ Build] から[Settings]を選択します。



- ③ [MCU Linker] から [Managed Linker Script] を選択し、[Link application to RAM] にチェックを入れます。

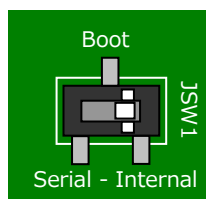


- ④ [Apply and Close] をクリックし、「3.1.3 ビルド方法」を参考に、プロジェクトをビルドしてください。



⑤ ボード上のディップスイッチを設定します。

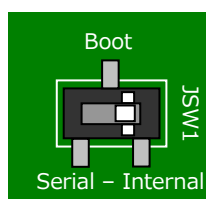
<NX-RT1021>



JSW1 : Internal

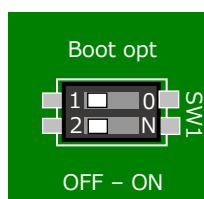
Internal Boot モード

<NX-RT1062>



JSW1 : Internal

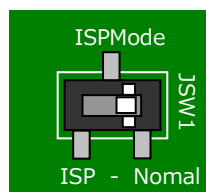
Internal Boot モード



SW1.1,1.2 : [OFF,OFF]

QSPI Flash Boot, Encrypted を使用しない

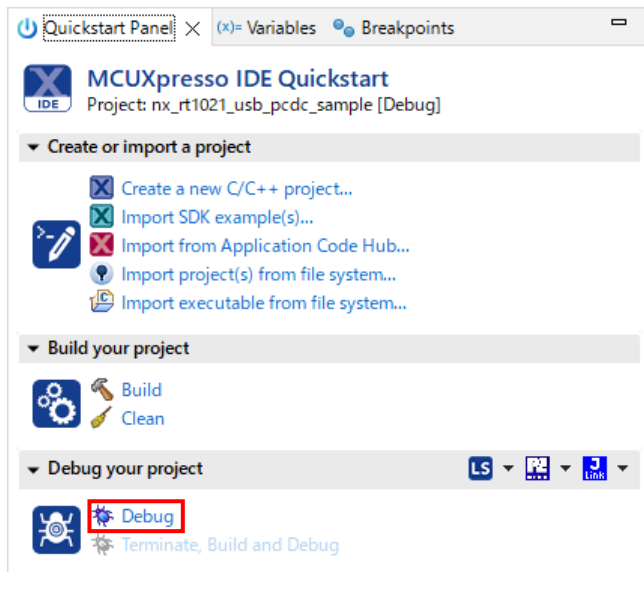
<NX-N947>



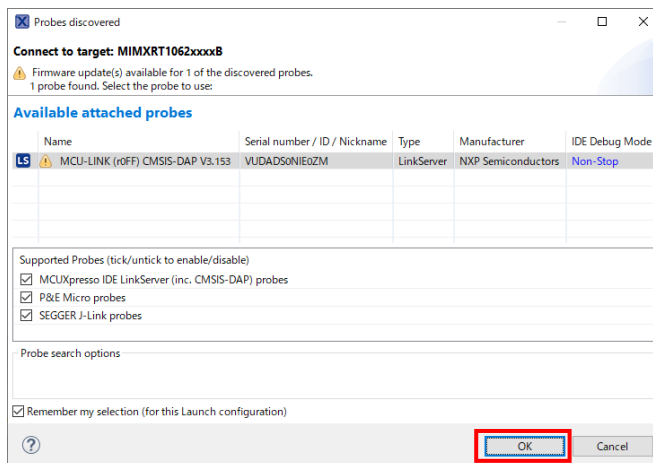
JSW1 : Nomal

Master Boot モード

- ⑥ ボードに電源を投入してください。
- ⑦ 画面左下の [Quick start Panel] にある [Debug your project] の [Debug] をクリックします。

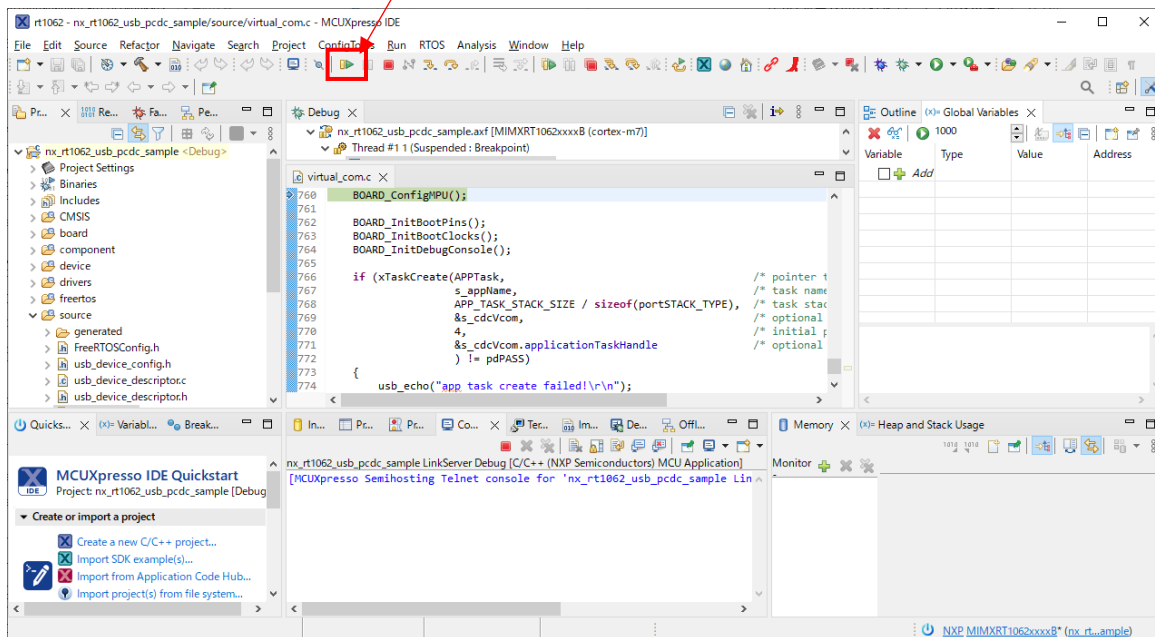


- ⑧ 下記の画面が表示された場合、「OK」をクリックしてください。



- ⑨ ボードとの接続が完了したら、[Resume]アイコンを押して、サンプルプログラムを実行させてください。

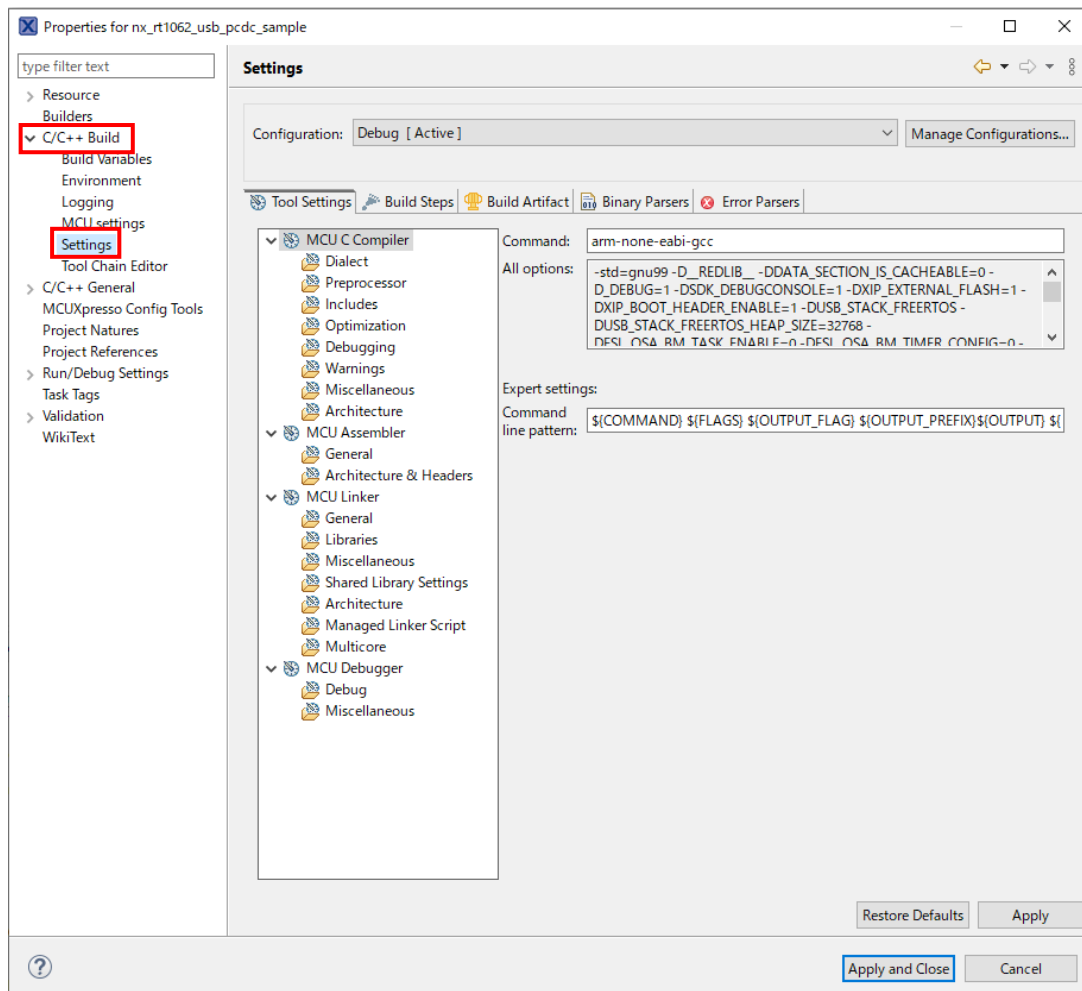
[Resume]アイコン



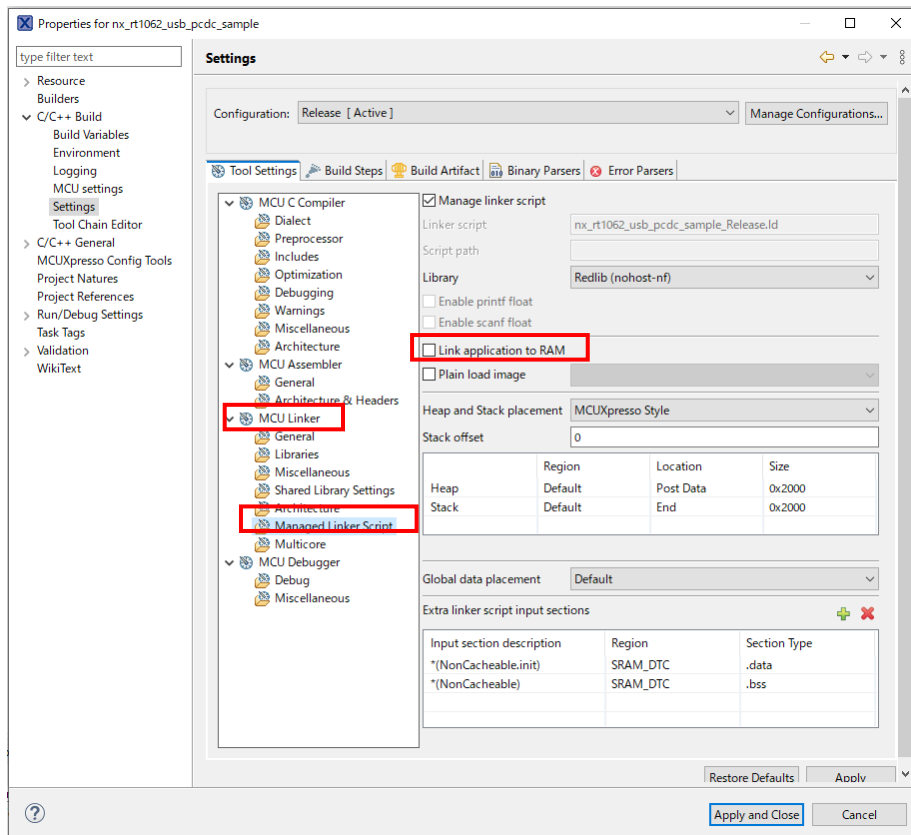
- ⑩ プログラムの動作が確認できましたら、CPU ボードへのプログラムのダウンロードも完了しています。

## 3.1.6 外部 ROM を用いたデバッグ、ダウンロード方法

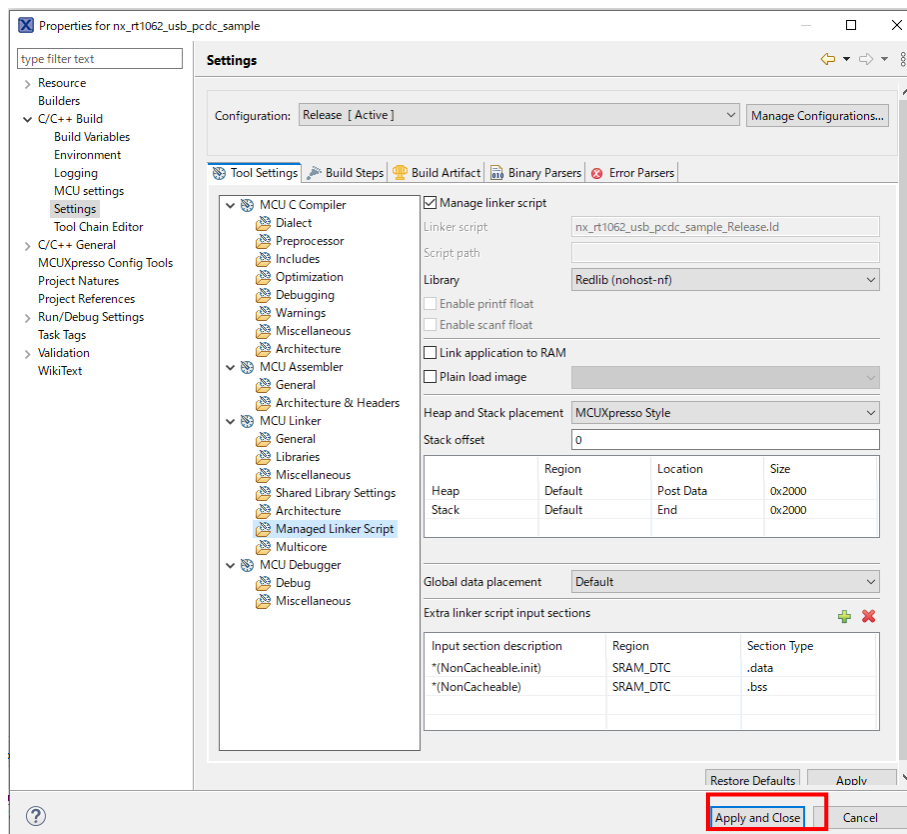
- ① ビルド構成に [Release] を選択します。
- ② Project Explorer にあるプロジェクトを右クリックし、[Properties]を開き、[C/C++ Build] から[Settings]を選択します。



- ③ [MCU Linker] から [Managed Linker Script] を選択し、[Link application to RAM] にチェックが入っていないことを確認します。

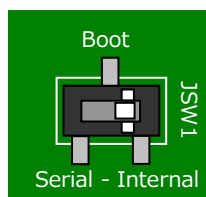


- ④ [Apply and Close] をクリックし、「3.1.3 ビルド方法」を参考に、プロジェクトをビルドしてください。



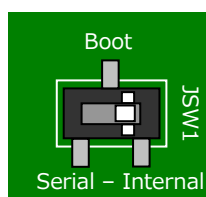
⑤ ボード上のディップスイッチを設定します。

<NX-RT1021>

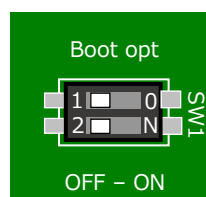


JSW1 : Internal  
Internal Boot モード

<NX-RT1062>

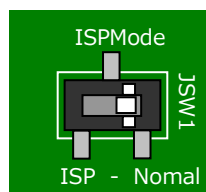


JSW1 : Internal  
Internal Boot モード



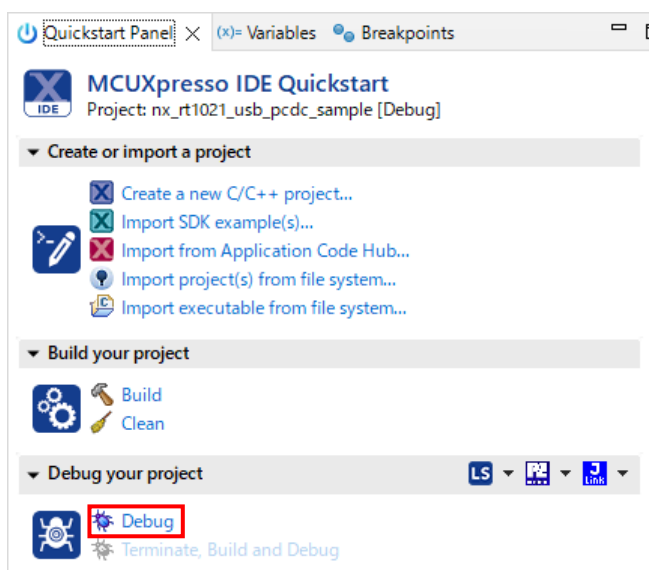
SW1.1,1.2 : [OFF,OFF]  
QSPI Flash Boot, Encrypted を使用しない

<NX-N947>

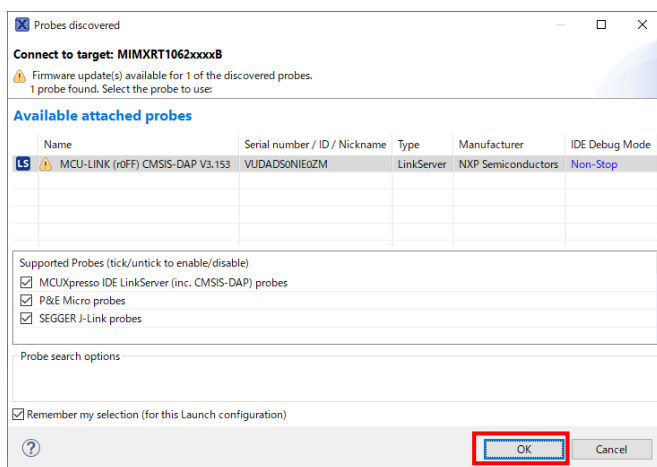


JSW1 : Nomal  
Master Boot モード

- ⑥ ボードに電源を投入してください。
- ⑦ 画面左下の [Quick start Panel] にある [Debug your project] の [Debug] をクリックします。

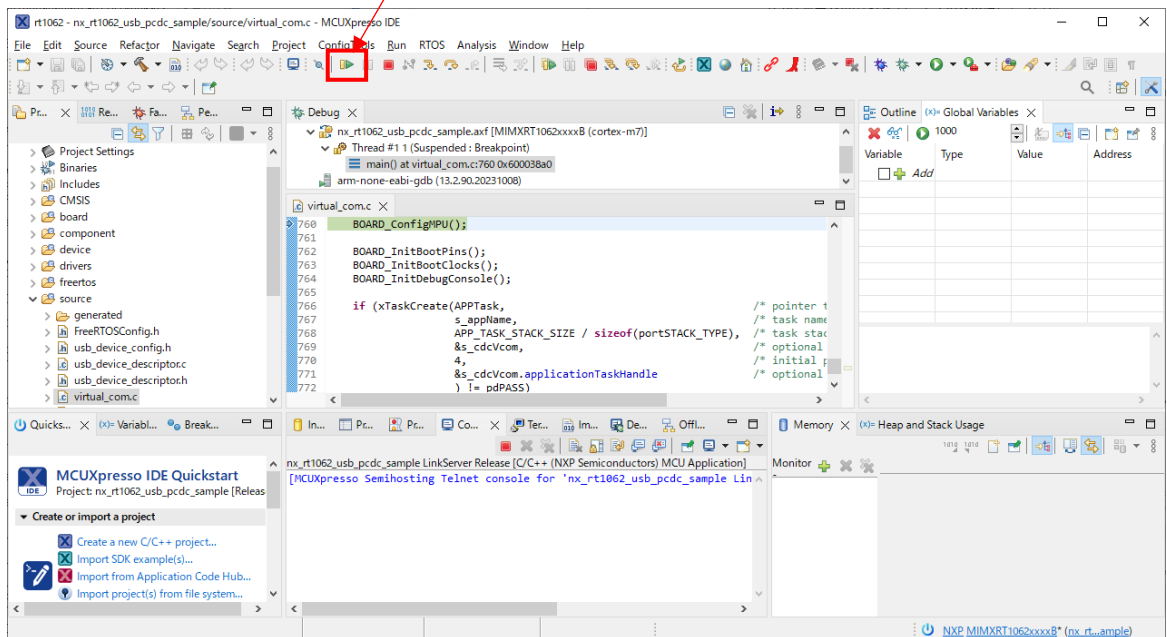


- ⑧ 下記の画面が表示された場合、「OK」をクリックしてください。



- ⑨ ボードとの接続が完了したら、[Resume]アイコンを押して、サンプルプログラムを実行させてください。

[Resume]アイコン

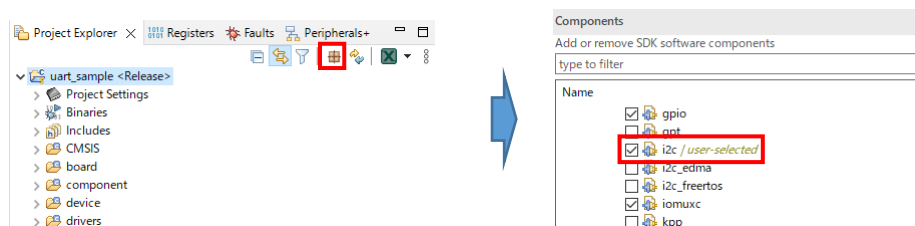


- ⑩ プログラムの動作が確認できましたら、CPU ボードへのプログラムのダウンロードも完了しています。弊社提供のサンプルプログラムは XIP モードを使用していますので、ダウンロードにて FlashROM へ書き込まれたプログラムは、電源投入または基板上的リセット SW 押下により動作します。

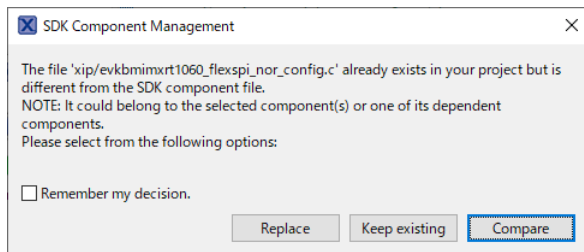
### 3.2 サンプルプログラムに機能を追加修正する際の注意事項

MCUXpresso では、機能の追加修正を[Manage SDK components]や[config ツール]によって行えます。機能追加の例として、uart\_sample へ I2C ドライバを追加する手順で説明します。修正方法と、ツールによるコード出力時の注意事項を記載します。

- ① [Manage SDK components]を開き、I2C のドライバを追加します。  
[Manage SDK components]でのドライバの追加方法は、4.3.1 項を参照してください。



- ② [Manage SDK components]の[OK]をクリックし、次に開く、ファイルの追加とアップデートを確認するダイアログが表示されますので[Yes]をクリックします。
- ③ 下記のダイアログが表示されます。  
これは、「xip/evkbmimxrt1060\_flexspi\_nor\_config.c」を弊社で修正しているため、[Manage SDK components]から提供されるファイルと違うことから書き換えの確認をするためのものです。  
書き換えを行わないために [Keep existing]をクリックします。 ([Replace]をクリックしないようご注意ください)



以上で、ドライバの追加が行えます。

- ③にて、「xip/evkbmimxrt1060\_flexspi\_nor\_config.c」の書き換えの確認時に[Replace]をクリックした場合、「xip/evkbmimxrt1060\_flexspi\_nor\_config.c」が書き換えられてしまいます。そのままビルドして使用するとサンプルプログラムが正しく動作してくれません。  
この場合、元のサンプルプログラムから「xip\evkbmimxrt1060\_flexspi\_nor\_config.c」をコピーすることで対応することができます。

その他にも弊社にて、[Manage SDK components]や[config ツール]から出力されるファイルに修正を加えているファイルがあります。

サンプルプログラムに機能追加を行い、正しく動作しなくなった場合には、[Manage SDK components]や[config ツール]から出力されたファイルを元のサンプルプログラムと比較し、必要な修正を行ってください。

## 4. MCUXpresso IDE を用いた新規プロジェクト作成方法

本章では、MCUXpresso IDE で CPU ボード用プログラムを作成する方法を説明します。

プロジェクトコンフィギュレータの設定を行うプロジェクトとして、「4.1 UART プログラムの作成例」にて、UART を使用したプログラムの作成方法を説明します。

なお、本書で扱っている MCUXpresso IDE のバージョンは V24.12[Build 148]です。異なるバージョンの MCUXpresso IDE をご使用の場合は、本書で解説する項目・設定等が異なる場合がございます。ご注意ください。

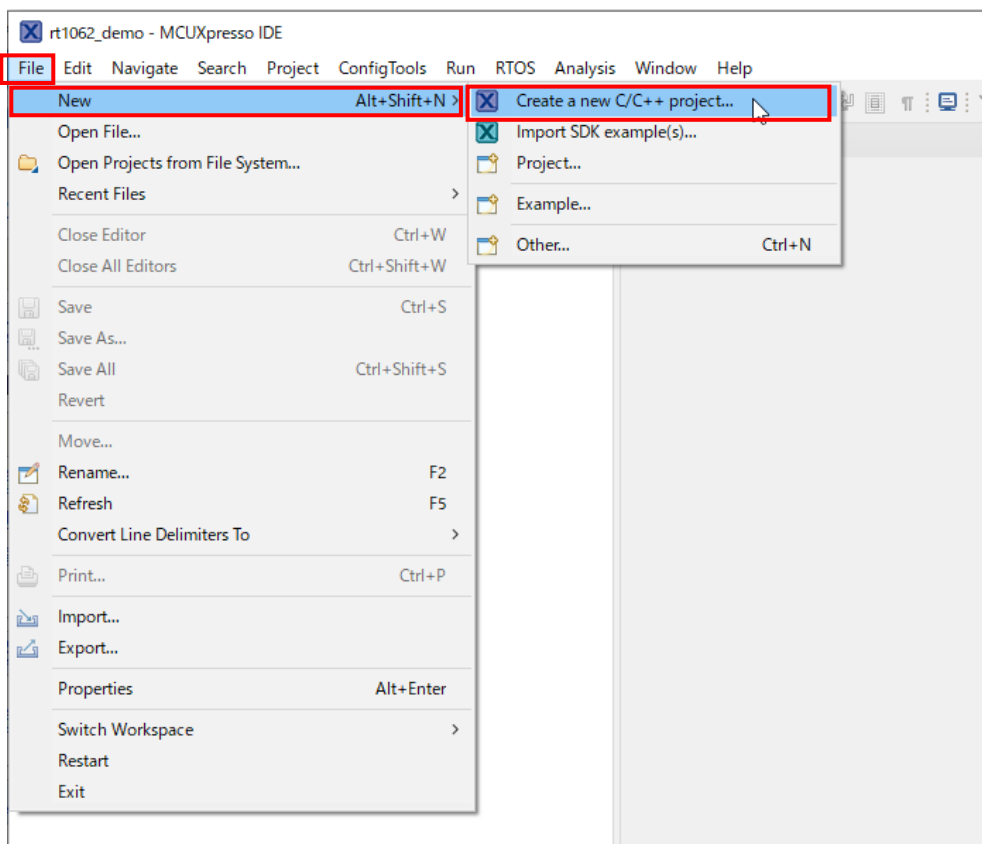
SDK に関してはインストール済みであるものとしますので、事前にインストールを行ってください。

また、本章では、図中の選択・実行する必要がある箇所を赤色枠 、内容の確認をする箇所を橙色枠  で示します。

### 4.1 内蔵 RAM 上で動作するプログラムの作成例

本節では、NX-RT1062 で動作する UART のプログラムを例に、プロジェクトの作成方法を説明します。

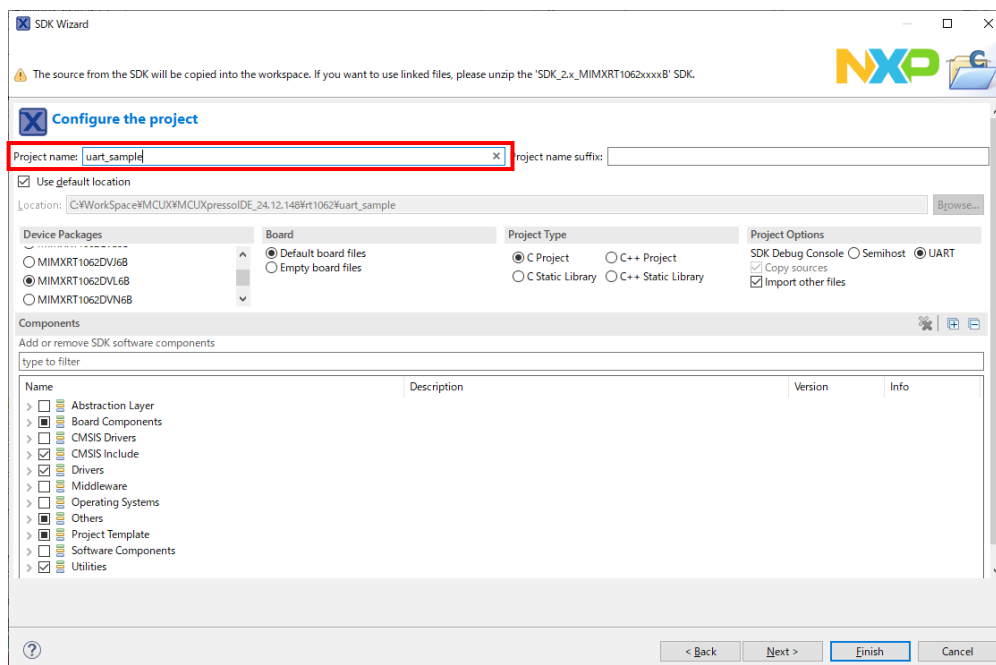
- ① MCUXpresso IDE のメニューから、[File] - [New] - [Create a new C/C++ Project]を選択します。



- ② [SDK MCUs] から使用する MCU を選択し、[Next>]を選択します。  
 NX-RT1021 の場合[evkmimxrt1020]を選択します。  
 NX-RT1062 の場合[evkbmimxrt1060]もしくは[evkcmimxrt1060]を選択します。  
 NX-N947 の場合[frdmmcxn947]を選択します。



- ③ [Project name] に任意のプロジェクト名を入力します（ここでは「uart\_sample」とします）。



## ④ NX-RT1021 の場合

[Device Packages]を[MIMXRT1021CAF4B]に設定します。

[Board]を[Default board files]に設定します。

[Project Type]を[C Project]に設定します。

[Project Options]の[SDK Debug Console]を[UART]に選択したうえで、[Import other files]にチェックを入れます。

[Components]から[Driver] - [Device] - [SDK Driver] - [lpuart]にチェックを入れ、[Next >]を選択します。

## NX-RT1062 の場合

[Device Packages]を[MIMXRT1062CVJ5B]に設定します。

[Board]を[Default board files]に設定します。

[Project Type]を[C Project]に設定します。

[Project Options]の[SDK Debug Console]を[UART]に選択したうえで、[Import other files]にチェックを入れます。

[Components]から[Driver] - [Device] - [SDK Driver] - [lpuart]にチェックを入れ、[Next >]を選択します。

## NX-N947 の場合

[Device Packages]を[MCCXN947VDF]に設定します。

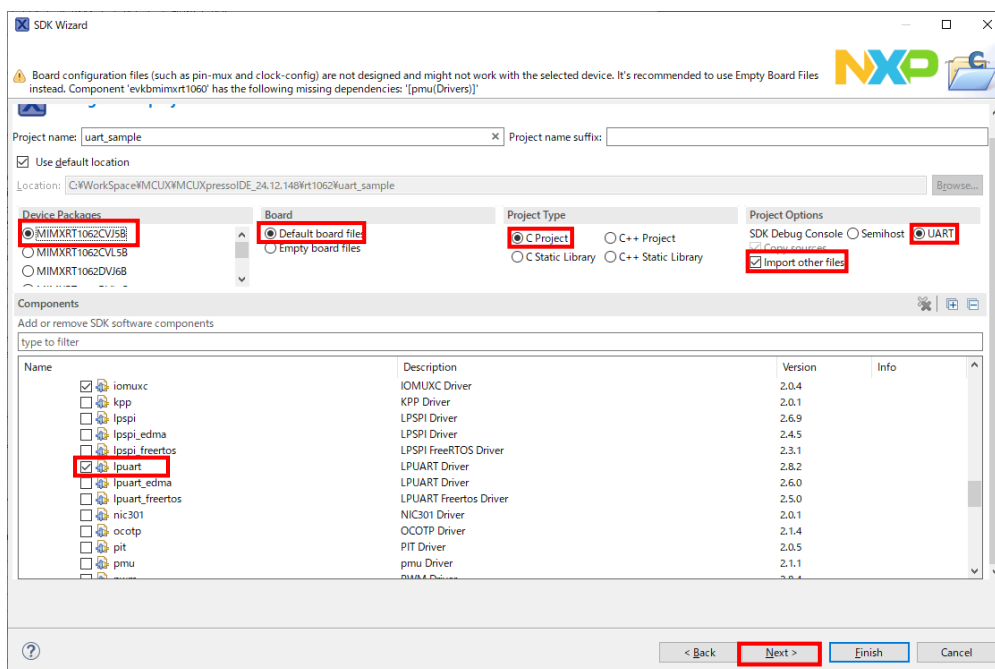
[Board]を[Default board files]に設定します。

[Project Type]を[C Project]に設定します。

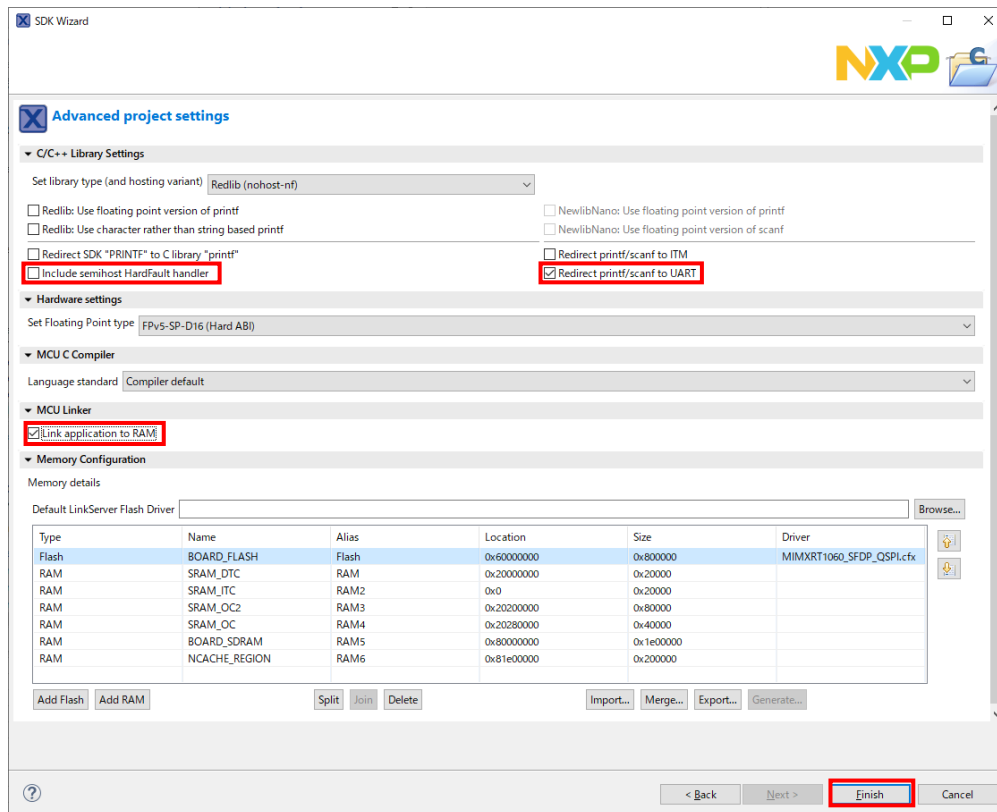
[Project Options]の[SDK Debug Console]を[UART]に選択したうえで、[Import other files]にチェックを入れます。

[Cores]から[cm33\_core0]を選択したうえで、[Role]を[Standalone]に選択します。

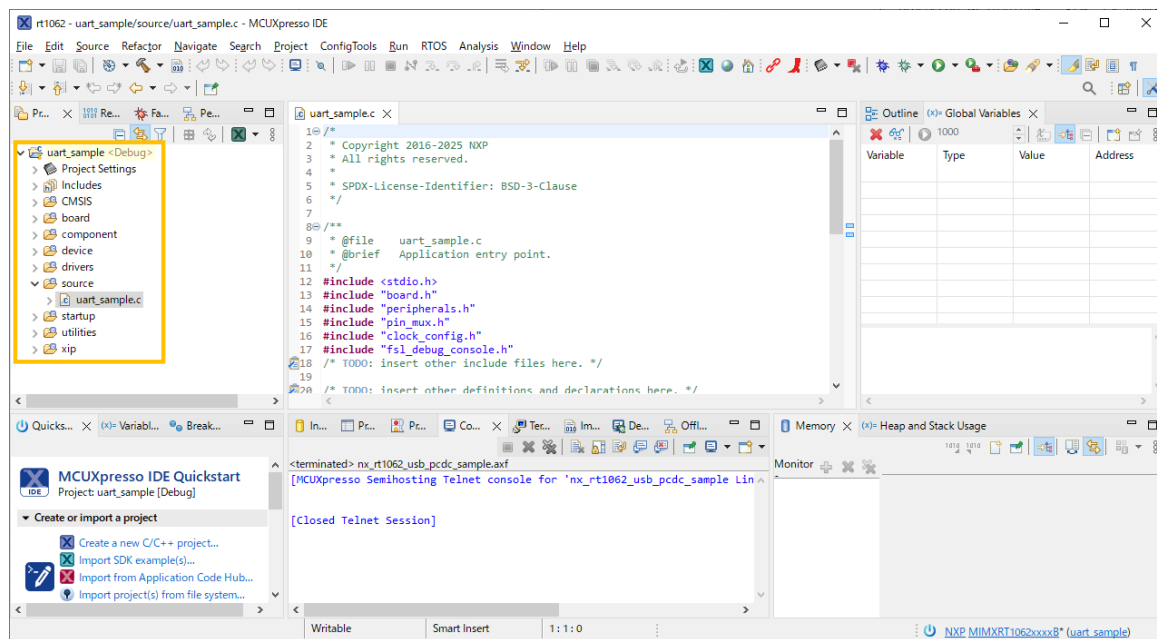
[Components]から[Driver] - [Device] - [SDK Driver] - [lpuart]にチェックを入れ、[Next >]を選択します。



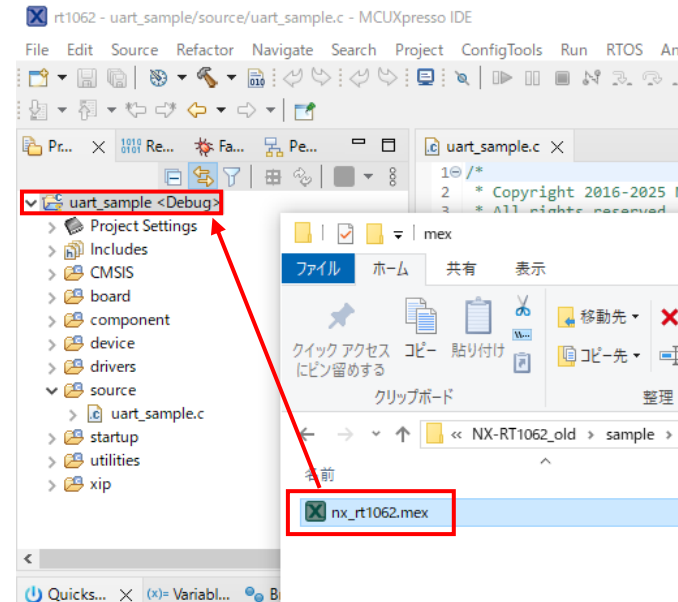
- ⑤ [Advanced project settings]を設定します。  
 [C/C++ Library Settings]の[Include semihost HardFault handler]のチェックを外し、  
 [Redirect printf/scanf to UART]にチェックを入れます。  
 [MCU Linker]の[Link application to RAM]にチェックを入れ、[Finish]を選択します。



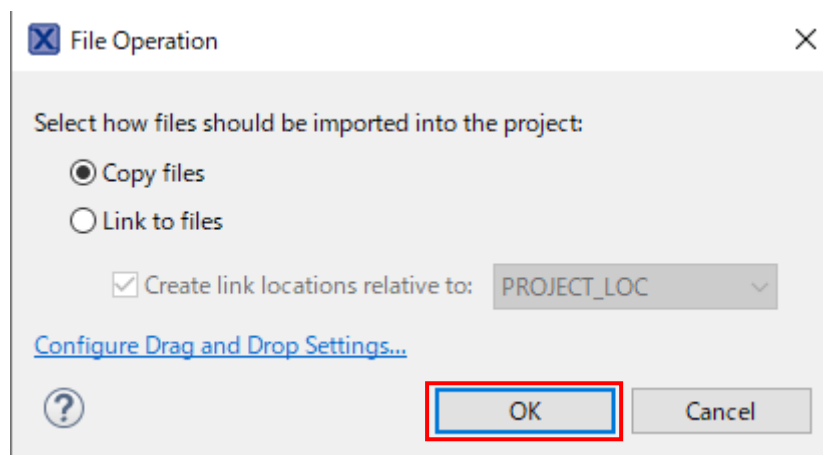
## ⑥ プロジェクトが作成されました。



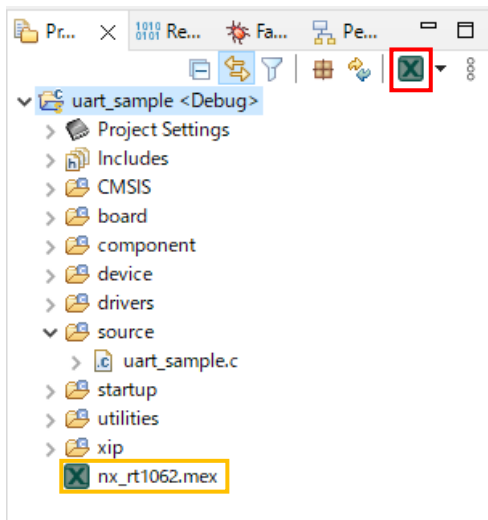
- ⑦ 弊社製 CPU ボード用に雛形としてクロックのみが設定されている“nx\_rt1062.mex”※(MCUXpresso Config Tools Settings File)を作成されたプロジェクトにドラッグアンドドロップします。
- ※ NX-RT1021 は、nx\_rt1021.mex、NX-N947 は、nx\_n947.mex を使用します。
- それぞれの mex ファイルは、サンプルプログラムを展開したフォルダ内の mex フォルダにあります。



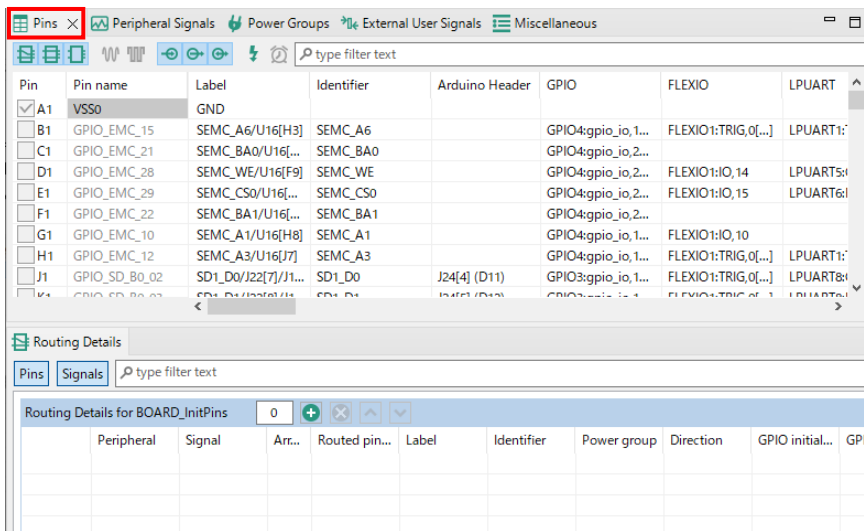
ドラッグアンドドロップした際に以下の画面が出ますが、[Copy files]を選択した状態で、[OK]を選択します。



- ⑧ プロジェクトに mex ファイルが反映されたことが確認できましたら、以下のアイコンをクリックし、Config ツールを開きます。



- ⑨ [Pins] タブを開きます。この画面で、Pin 設定をします。



⑩ 設定するピンは、ボード毎に下記の UART 端子を設定します。

NX-RT1021 の場合

- Pin name [GPIO\_AD\_B0\_06]に[LPUART1:TX]を選択
- Pin name [GPIO\_AD\_B0\_07]に[LPUART1:RX]を選択

NX-RT1062 の場合

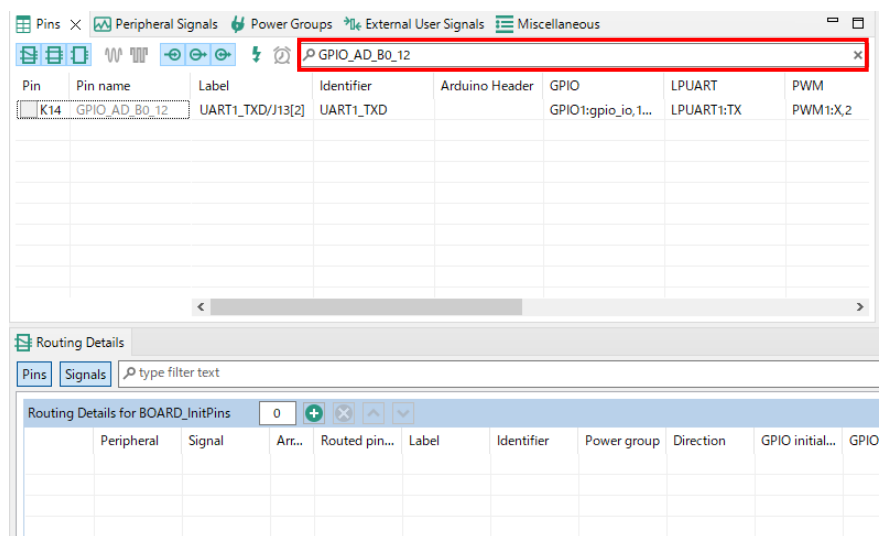
- Pin name [GPIO\_AD\_B0\_12]に[LPUART1:TX]を選択
- Pin name [GPIO\_AD\_B0\_13]に[LPUART1:RX]を選択

NX-N947 の場合

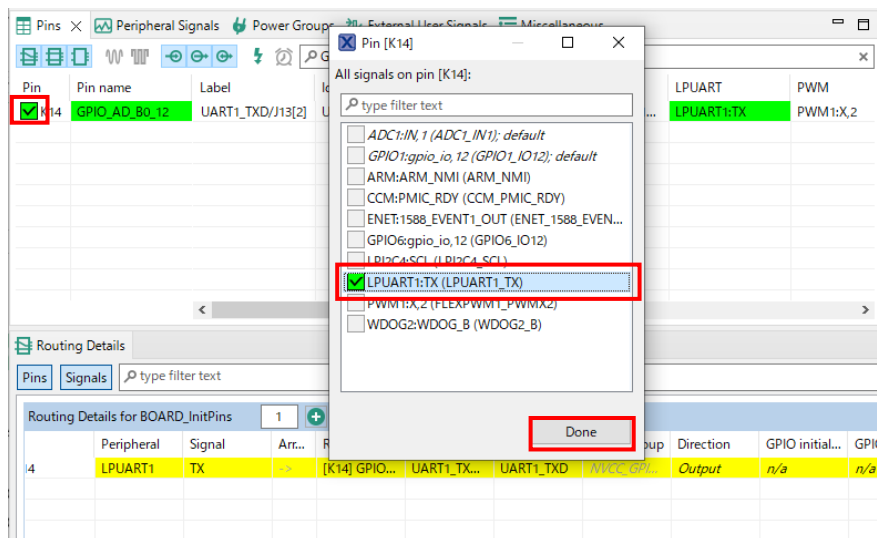
- Pin name [PIO1\_9]に[LPFLEXCOMM4:P1,SCK/TXD/SCL]を選択
- Pin name [PIO1\_8]に[LPFLEXCOMM4:P0,SDO/RXD/SDA]を選択

⑪ UART の送信端子 (TX/TXD) を設定します。

送信端子の[Pin name]を[type filter text]に入力すると下記のようにその端子のみ表示されます。



⑫ [Pin]の欄のボックスをクリックし、表示されたダイアログから送信端子を選択し、[Done]をクリックします。



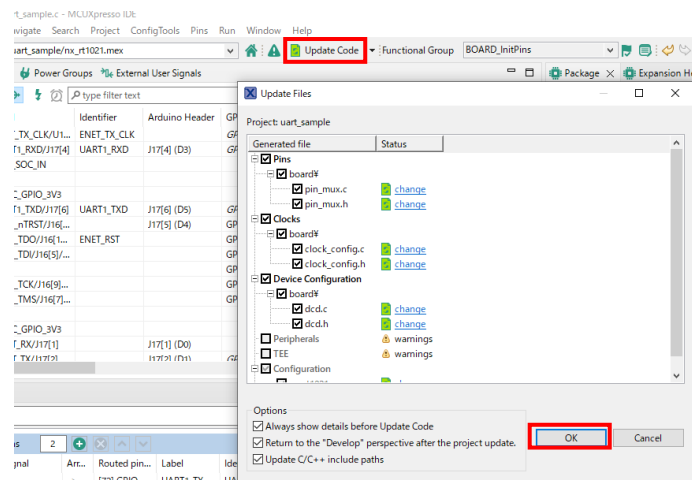
- ⑬ 選択された端子は、[Routing Details]に表示されます。

| #   | Peripheral | Signal | Arr... | Routed pin... | Label       | Identifier | Power group | Direction | GPIO initial... | GPIO interr... | Software In... |
|-----|------------|--------|--------|---------------|-------------|------------|-------------|-----------|-----------------|----------------|----------------|
| K14 | LPUART1    | TX     | ->     | [K14] GPIO... | UART1_TX... | UART1_TXD  | NVCC_GPI... | Output    | n/a             | n/a            | Disabled       |

- ⑭ 同様に UART の受信端子 (RX/RXD) の設定を行い、[Routing Details]に受信端子が表示されることを確認します。

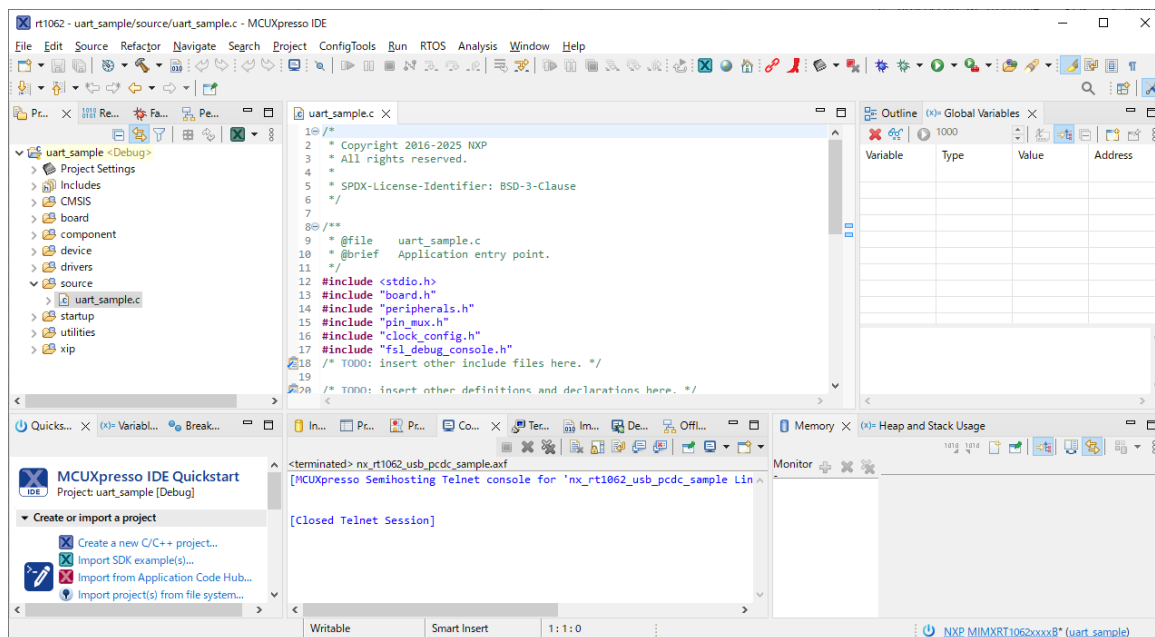
| #   | Peripheral | Signal | Arr... | Routed pin... | Label       | Identifier | Power group | Direction | GPIO initial... | GPIO interr... | Software In... |
|-----|------------|--------|--------|---------------|-------------|------------|-------------|-----------|-----------------|----------------|----------------|
| L14 | LPUART1    | RX     | <-     | [L14] GPIO... | UART1_RX... | UART1_RXD  | NVCC_GPI... | Input     | n/a             | n/a            | Disabled       |

- ⑮ Pin の設定完了後、完了した値を反映させるため[Update Code]を選択してください。  
以下の画面が出るため[OK]を選択してください。



今回は、UART のみの Pin 設定でしたが、続けてほかのモジュールの Pin 設定を行う場合もあります。  
その都度、必要な設定を行ってください。

- ⑩ 以上で UART のプロジェクトが完成しました。



- ⑪ ビルドを行い、PC とボードの UART のコネクタを PC-USB-04 等で接続し、ターミナルソフトを起動した状態で、デバッガを起動し、実行してください。  
ターミナルソフトに「Hello World」と表示されれば、作成したプログラムは動作しています。

## 4.2 外部 ROM 上で動作するプログラムの作成例

XIP モードで動作する外部の FlashROM 上で動作するプログラムは、XIP モードにて動作します。

XIP モードでは MCUXpresso IDE からのダウンロードで書き込みが行え、そのままデバッグをすることが可能です。

以下、作成したプロジェクトを XIP に対応させる方法について説明します。

### 4.2.1 NX-RT1021、NX-RT1062 の変更内容

#### ① XIP の設定ファイルの変更

XIP の設定は、下記の表にあるファイルにて行っております。

弊社提供の各サンプルプログラムの下記にあるファイルを作成したプロジェクトのファイルと差し替えてください。

ファイルの内容につきましては、各リファレンスマニュアルの「9.6.3 Serial NOR and NAND Configuration based on FlexSPI Interface」を参照してください。

| ボード       | ファイル                                      |
|-----------|---|
| NX-RT1021 | ¥xip¥evkmimxrt1020_flexspi_nor_config.c   |
| NX-RT1062 | ¥xip¥evkmbmimxrt1060_flexspi_nor_config.c |

#### ② リンカスクリプトの設定

「3.1.5 外部 ROM を用いたデバッグ、ダウンロード方法」の②～⑤を参考に Link application to RAM] にチェックを外します。

以上で、NX-RT1021、NX-RT1062 の XIP の設定は、終了です。

ビルドを行って、デバッグにて、ダウンロードし、動作を確認してください。

### 4.2.2 NX-N947 の変更内容

#### ① XIP の設定ファイルの変更

XIP の設定は、下記の表にあるファイルにて行っております。

弊社提供の各サンプルプログラムの下記にあるファイルを作成したプロジェクトのファイルと差し替えます。

ファイルの内容につきましては、MCX Nx4x リファレンスマニュアルの「15.3.1.2 FlexSPI NOR flash memory boot」を参照してください。

| ボード     | ファイル                               |
|---------|------------------------------------|
| NX-N947 | ¥source¥mccxn_flexspi_nor_config.c |

#### ② リンカスクリプトの設定 1

「3.1.5 外部 ROM を用いたデバッグ、ダウンロード方法」の②～⑤を参考に Link application to RAM] にチェックを外します。

#### ③ リンカスクリプトの設定 2

弊社提供の各サンプルプログラムの「¥linkscripts」フォルダを作成したプロジェクトへコピーします。

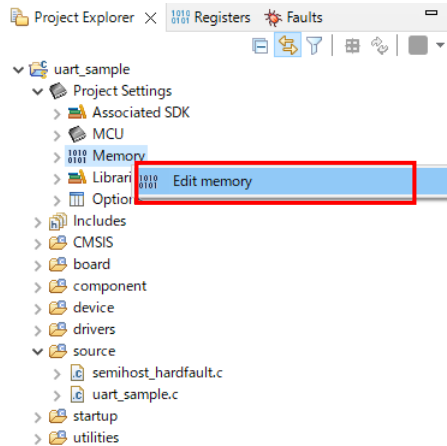
#### ④ Flash Driver ファイルの追加

MCUXpresso IDE をインストール時に指定したフォルダ内にある「¥LinkServer\_24.12.21¥binaries¥Flash」へ弊社提供の NX-N947 のサンプルプログラムを展開したフォルダ内「xip」フォルダにある「MCXN9xx\_SFDP\_FlexSPI\_NX\_N947.cfx」をコピーします。

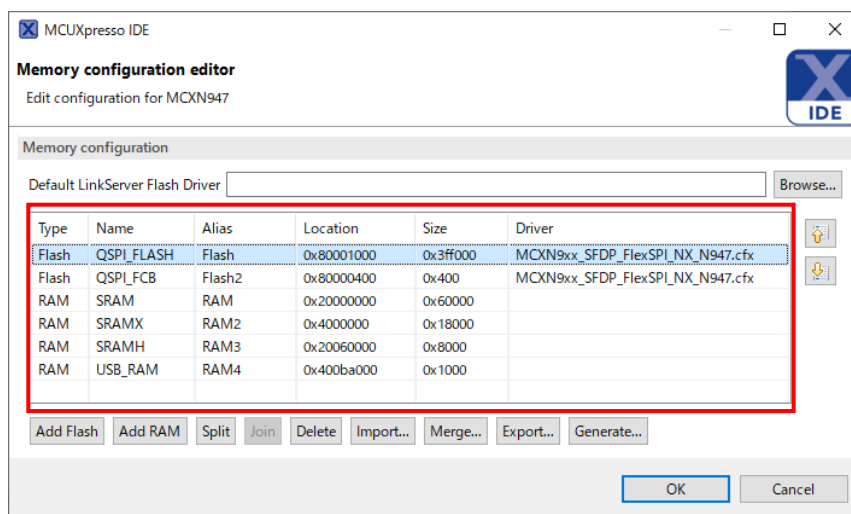
## ⑤ メモリ設定

メモリ設定に FlashROM (QSPI Flash) の登録を行います。

「Project Explorer」のプロジェクトの中にある「Project Setting」-「Memory」を右クリックし、表示された「Edit memory」をクリックします。



表示された「Memory configuration editor」の設定値を下記の赤枠の中の内容に変更してください。



以上で、NX-N947 の XIP の設定は、終了です。

ビルドを行って、デバッガにて、ダウンロードし、動作を確認してください。

その他、NX-N947 で XIP を使用した場合の注意事項につきましては、「AN2305 NX-N947 サンプルプログラム解説」を参照してください。

### 4.3 SDRAM を使用するプログラムの作成例

NX-RT1062 は、SDRAM が実装されています。

SDRAM を動作させるためには、SEMC (Smart External Memory Controller) の設定やアドレスバス、データバス、RAS、CAS 等信号線のポート設定を行う必要があります。

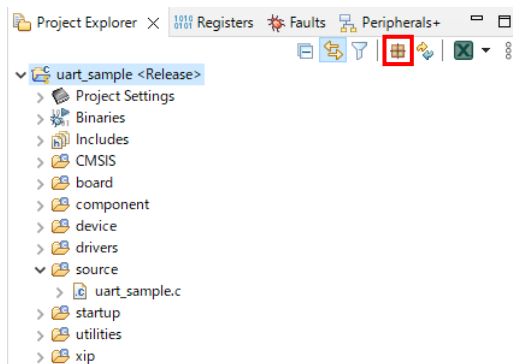
また、FlashROM から起動する場合には、DCD (Device Configuration Data) を使用し、SDRAM の設定を行う必要があります。

なお、SDRAM を使用する場合の DCD の設定につきましては、「nx\_rt1062.mex」にて、行っております。

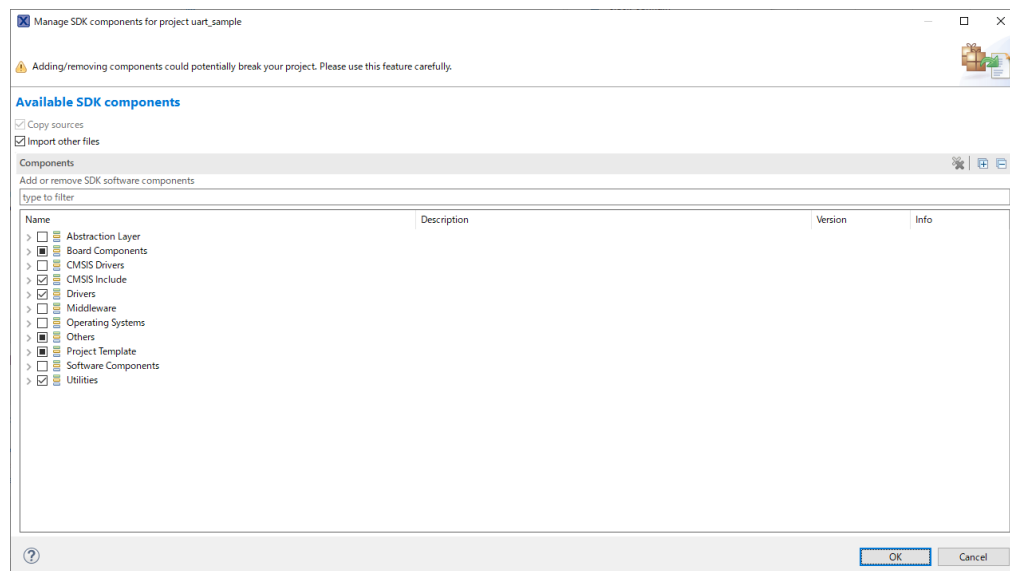
以下、SDRAM の設定方法を説明します。

#### 4.3.1 SEMC の設定

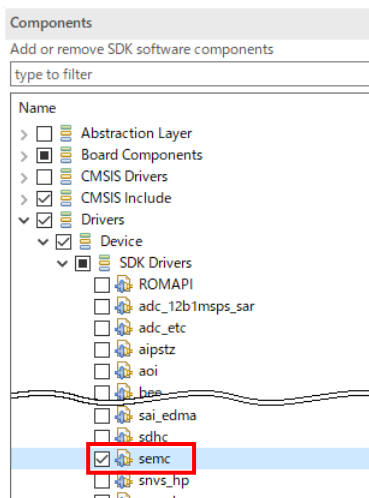
- ① [Manage SDK components]アイコンをクリックし、[Manage SDK components]を開きます。



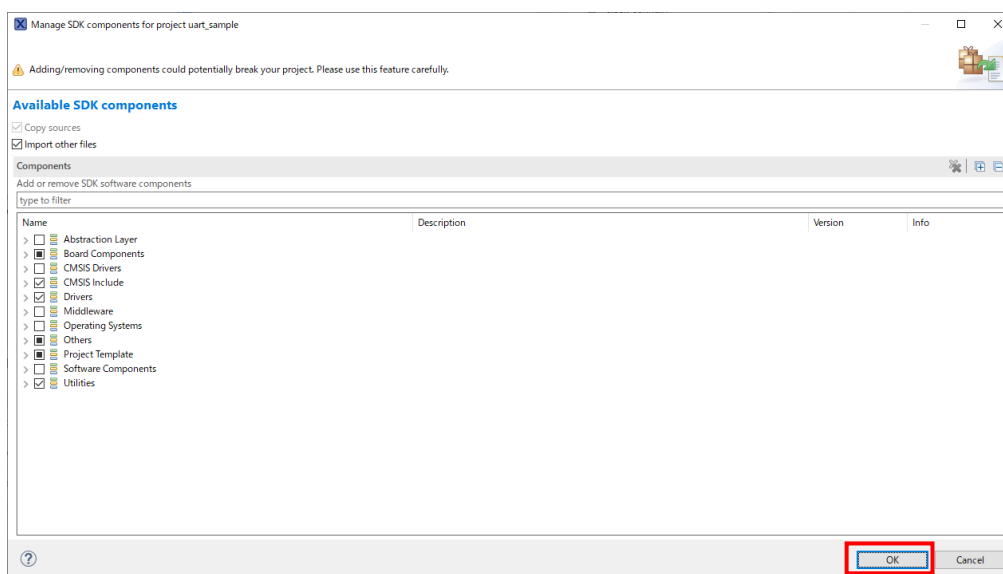
下記が、[Manage SDK components]となります。



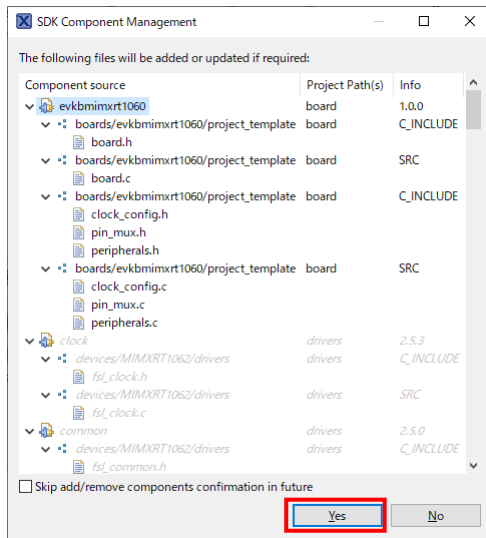
- ② [Components]の[Drivers]-[Device]-[SDK Drivers]にある[semc]のチェックを入れます。



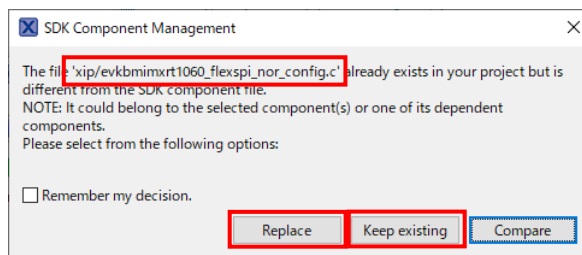
- ③ [OK]をクリックします。



- ④ ファイルの追加とアップデートを確認するダイアログが表示されますので[Yes]をクリックします。



- ⑤ 更新するファイル毎に更新を行うか確認するダイアログが表示されます。  
「xip/evkbmimxrt1060\_flexspi\_nor\_config.c」の更新のときのみ[Keep existing]を、それ以外は[Replace]をクリックします。  
もし、「xip/evkbmimxrt1060\_flexspi\_nor\_config.c」を更新してしまったときは、「4.2.1」の「①」を参考にファイルを差し替えてください。



- ⑥ uart\_sample.c に SDRAM の設定を行うためのコードを追加します。

- (1) 「fsl\_semc.h」のインクルードを追記します。

```
#include "clock_config.h"
#include "fsl_debug_console.h"
#include "fsl_semc.h"

/* TODO: insert other include files here. */
```

「fsl\_semc.h」のインクルードを追記

- (2) 「EXAMPLE\_SEMC\_CLK\_FREQ」の宣言を追記します。

```
#include "clock_config.h"
#include "fsl_debug_console.h"
#include "fsl_semc.h"

/* TODO: insert other include files here. */
```

「EXAMPLE\_SEMC\_CLK\_FREQ」の  
宣言を追記

- (3) UART のサンプルプログラム（nx\_rt1062\_uart\_sample）の「source\uart\_interrupt.c」内にある「BOARD\_InitSEMC」関数を「main」関数の前に追加

```
/*
 * @brief Application entry point.
 */
status_t BOARD_InitSEMC(void)
{
    :
    :
}

int main(void) {
```

「BOARD\_InitSEMC」関数を追加

- (4) [main]関数の[BOARD\_InitBootPeripherals]関数の後に[BOARD\_InitSEMC]関数のコールを追記します。

```
int main(void) {
    unsigned int l;

    /* Init board hardware. */
    BOARD_ConfigMPU();
    BOARD_InitBootPins();
    BOARD_InitBootClocks();
    BOARD_InitBootPeripherals();
    BOARD_InitSEMC();
#ifdef BOARD_INIT_DEBUG_CONSOLE_PERIPHERAL
    /* Init FSL debug console. */
    BOARD_InitDebugConsole();
#endif
```

BOARD\_InitSEMC 関数のコールを追記

以上で、SDRAM の設定が行えるようになります。

## 4.3.2 ポートの設定

「4.1 内蔵 RAM 上で動作するプログラムの作成例」の⑧～⑭を参考に下記の端子設定を行います。

| Pin name    | 機能           |
|-------------|--------------|
| GPIO_EMC_09 | SEMC:ADDR,00 |
| GPIO_EMC_10 | SEMC:ADDR,01 |
| GPIO_EMC_11 | SEMC:ADDR,02 |
| GPIO_EMC_12 | SEMC:ADDR,03 |
| GPIO_EMC_13 | SEMC:ADDR,04 |
| GPIO_EMC_14 | SEMC:ADDR,05 |
| GPIO_EMC_15 | SEMC:ADDR,06 |
| GPIO_EMC_16 | SEMC:ADDR,07 |
| GPIO_EMC_17 | SEMC:ADDR,08 |
| GPIO_EMC_18 | SEMC:ADDR,09 |
| GPIO_EMC_23 | SEMC:ADDR,10 |
| GPIO_EMC_19 | SEMC:ADDR,11 |
| GPIO_EMC_20 | SEMC:ADDR,12 |
| GPIO_EMC_21 | SEMC:BA,0    |
| GPIO_EMC_22 | SEMC:BA,1    |
| GPIO_EMC_24 | SEMC:CAS     |
| GPIO_EMC_25 | SEMC:RAS     |
| GPIO_EMC_27 | SEMC:CKE     |
| GPIO_EMC_26 | SEMC:CLK     |
| GPIO_EMC_29 | SEMC:CS,0    |

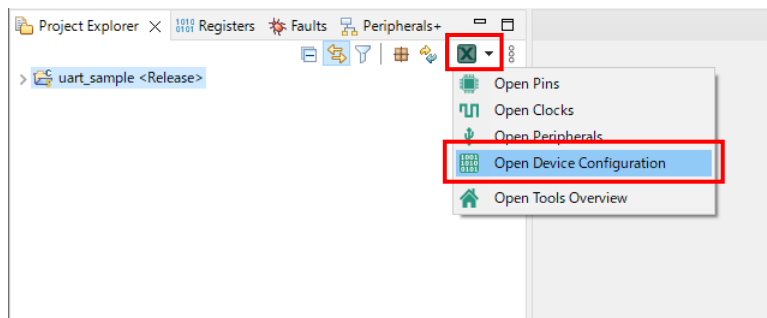
| Pin name    | 機能           |
|-------------|--------------|
| GPIO_EMC_00 | SEMC:DATA,00 |
| GPIO_EMC_01 | SEMC:DATA,01 |
| GPIO_EMC_02 | SEMC:DATA,02 |
| GPIO_EMC_03 | SEMC:DATA,03 |
| GPIO_EMC_04 | SEMC:DATA,04 |
| GPIO_EMC_05 | SEMC:DATA,05 |
| GPIO_EMC_06 | SEMC:DATA,06 |
| GPIO_EMC_07 | SEMC:DATA,07 |
| GPIO_EMC_30 | SEMC:DATA,08 |
| GPIO_EMC_31 | SEMC:DATA,09 |
| GPIO_EMC_32 | SEMC:DATA,10 |
| GPIO_EMC_33 | SEMC:DATA,11 |
| GPIO_EMC_34 | SEMC:DATA,12 |
| GPIO_EMC_35 | SEMC:DATA,13 |
| GPIO_EMC_36 | SEMC:DATA,14 |
| GPIO_EMC_37 | SEMC:DATA,15 |
| GPIO_EMC_08 | SEMC:DM,0    |
| GPIO_EMC_38 | SEMC:DM,1    |
| GPIO_EMC_28 | SEMC:WE      |

## 4.3.3 DCD の設定

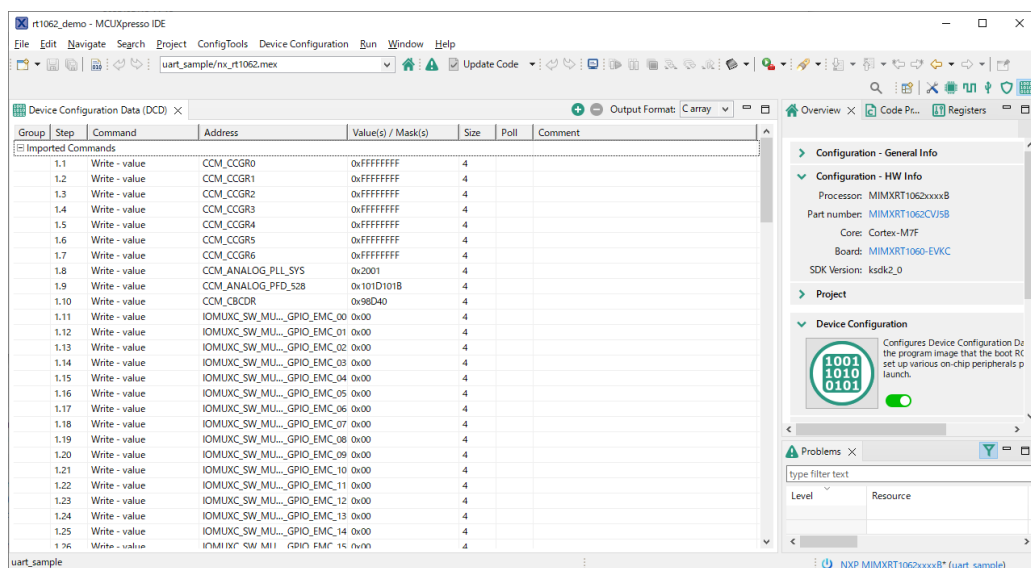
「nx\_rt1062.mex」を使用してプロジェクトを作成した場合、DCD については、設定を変更する必要はありませんが、DCD の追加、変更方法を説明します。

## ① DCD の設定画面を開く

DCD の設定を行うため、[Open MCUXpresso Configuration]アイコンの右にある[▼]をクリックし、表示されたプルダウンメニューの[Open Device Configuration]をクリックします。



[Device Configuration Data(DCD)]タグが開きます。

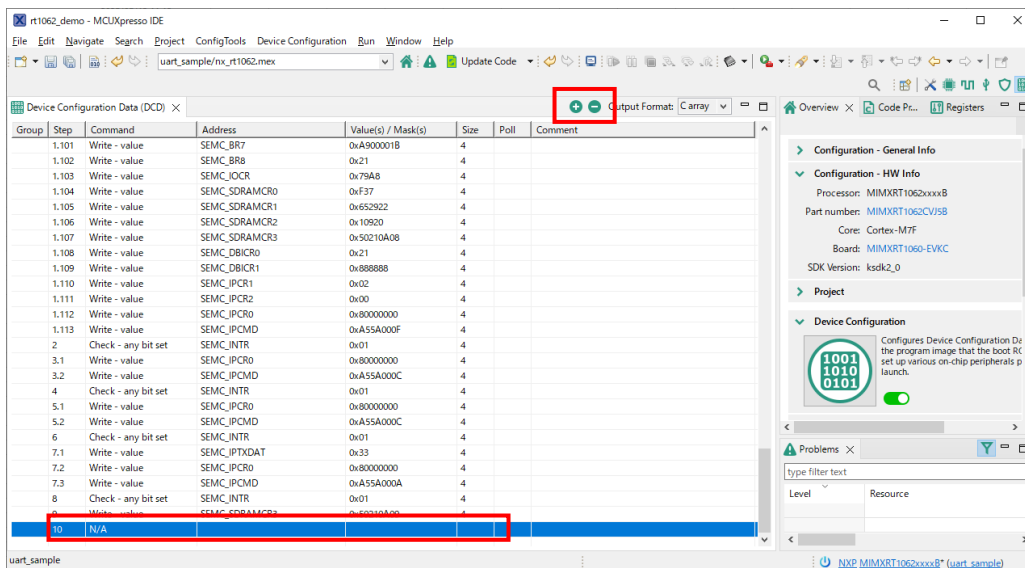




③ DCD の設定追加と削除

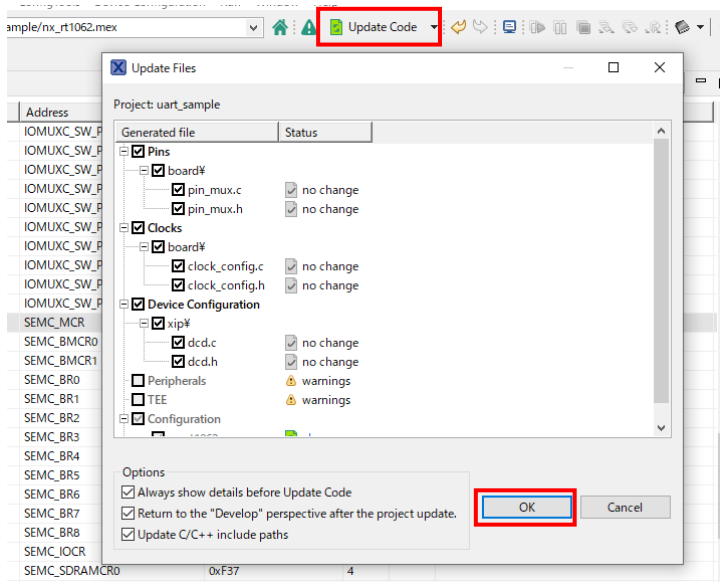
DCD の設定に新しい行を追加する場合には、追加したい行にカーソルを合わせ、[+]のアイコンをクリックすると新しい行が追加されます。

既存の行を削除する場合には、カーソルを削除したい行に合わせ、[-]のアイコンをクリックするとその行が削除されます。下記は、最後の行にカーソルを合わせ、[+]アイコンをクリックしたときのもので、[Step]の 10 に新しい行が追加されています。



④ DCD の設定終了

設定完了後、完了した値を反映させるため[Update Code]を選択してください。以下の画面が出るため[OK]を選択してください。



上記の処理後、ビルドを行うことで、SDRAM へのアクセスが行えるようになります。

## 5. MCUXpresso Secure Provisioning を使用した外部 FlashROM への書き込み

本章では、CPU ボードに MCUXpresso Secure Provisioning (以降、SEC ツールと記述します) を使用して、外部 QSPI Flash へのプログラムの書き込み方法を説明します。

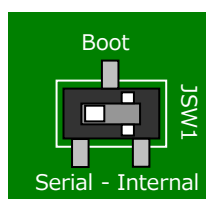
CPU 内蔵 ROM ですが、i.MX RT1021、i.MX RT1062 の内蔵 ROM へは、プログラムの書き込みは行えません。

MCX N947 の内蔵 FlashROM への書き込み方法につきましては、「MCUXpresso Secure Provisioning Tool User Guide」を参照してください。

### 5.1 USB インタフェースを使用した書き込み方法

- ① CPU ボードの設定を Serial Downloader モード、ISP モードに変更し、CPU ボードと PC を USB ケーブルで接続します。

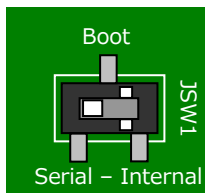
<NX-RT1021>



JSW1 : Serial

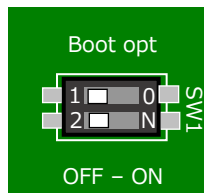
Serial Downloader モード

<NX-RT1062>



JSW1 : Serial

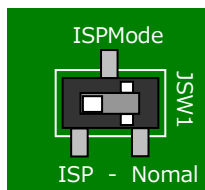
Serial Downloader モード



SW1.1,1.2 : [OFF,OFF]

QSPI Flash Boot, Encrypted を使用しない

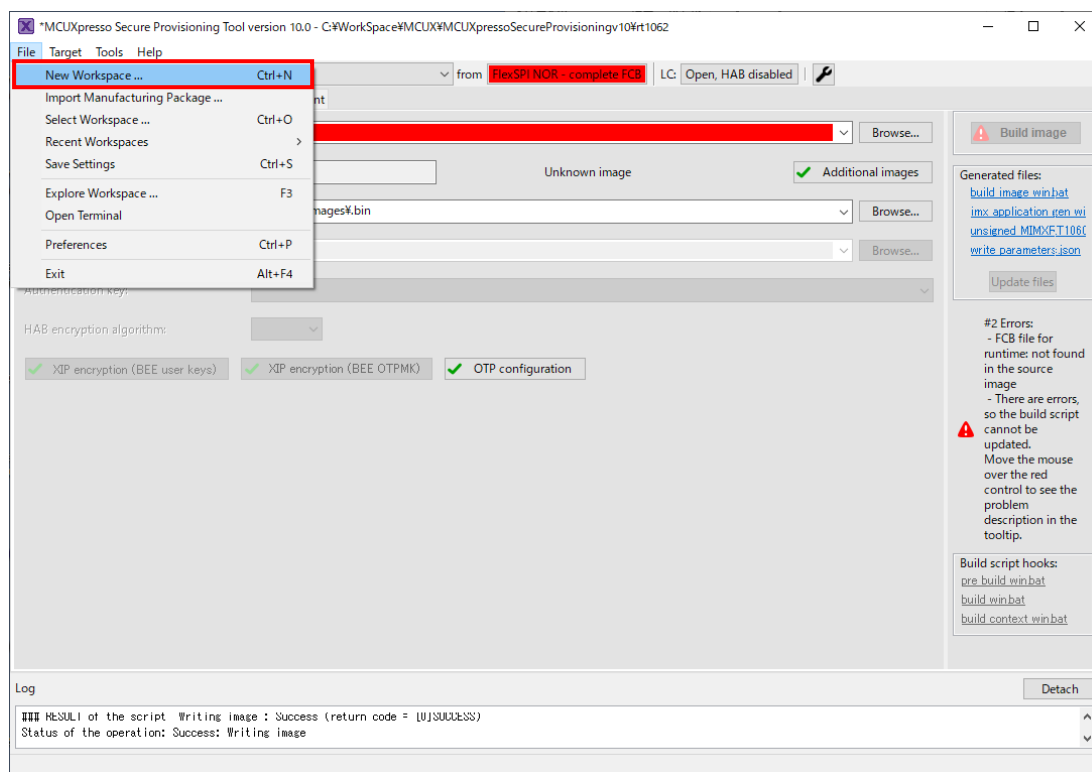
<NX-N947>



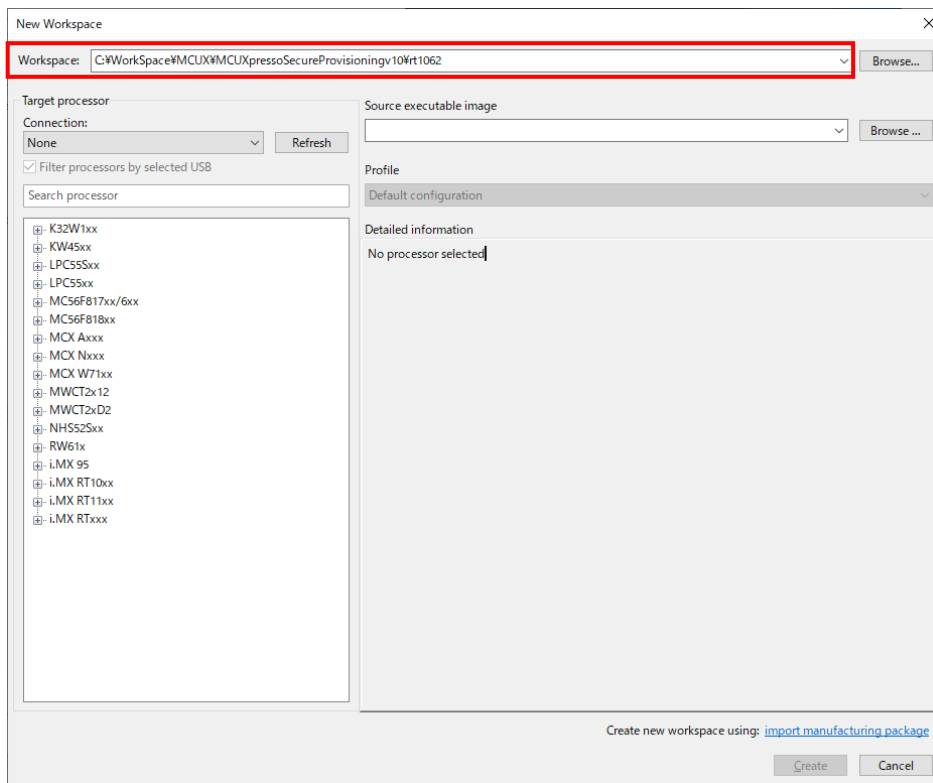
JSW1 : ISP

ISP モード

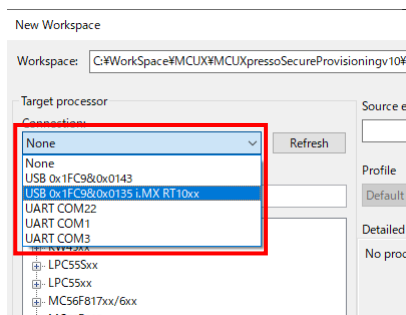
- ② CPU ボードの電源を入れます。
- ③ SEC ツールを起動します。
- ④ 2 回目以降に起動された場合は、メニューの [File] - [New Workspace...] をクリックします。  
始めて起動された場合は、⑤へ進んでください。



- ⑤ Workspace の欄にフルパスで Workspace 名を入力します。(フォルダの選択には、[Browse...] を使用してください)

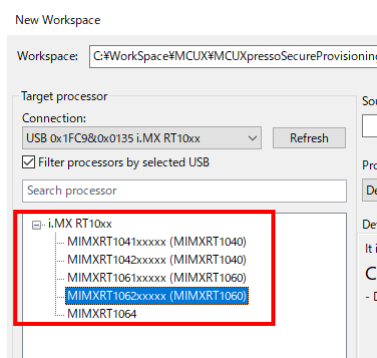


- ⑥ Connection からボードの USB ポートを選択します。



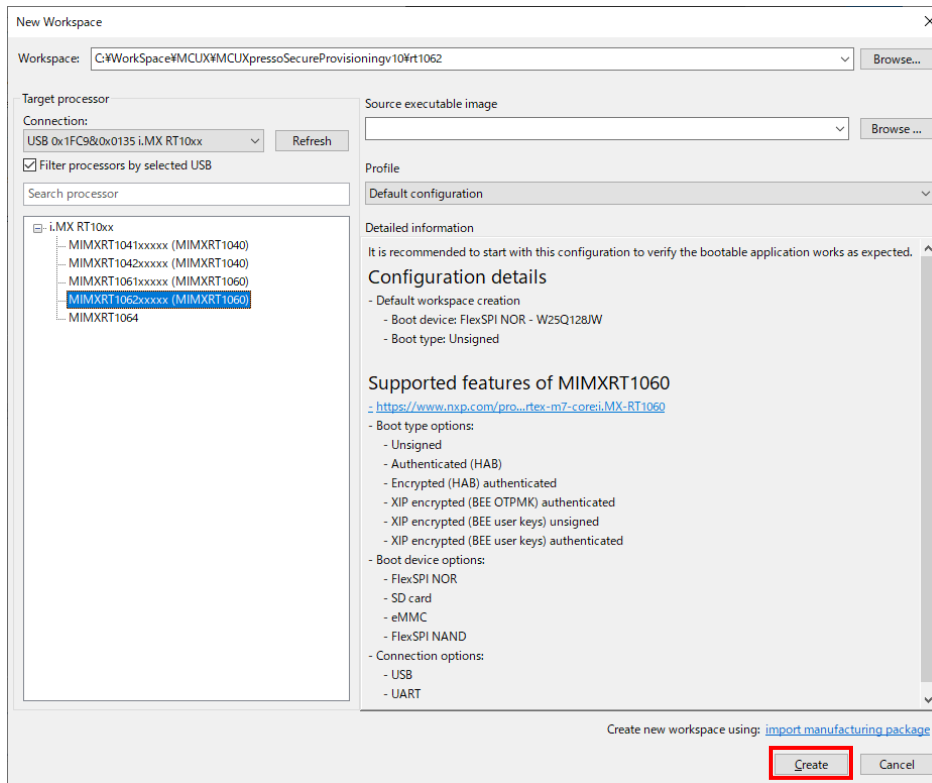
| ボード       | 選択するポート                       |
|-----------|-------------------------------|
| NX-RT1021 | USB 0x1FC9&0x0130 i.MX RT10xx |
| NX-RT1062 | USB 0x1FC9&0x0135 i.MX RT10xx |
| NX-N947   | USB 0x1FC9&0x014F Nxxx        |

- ⑦ Connection の下にあるツリーから CPU を選択します。

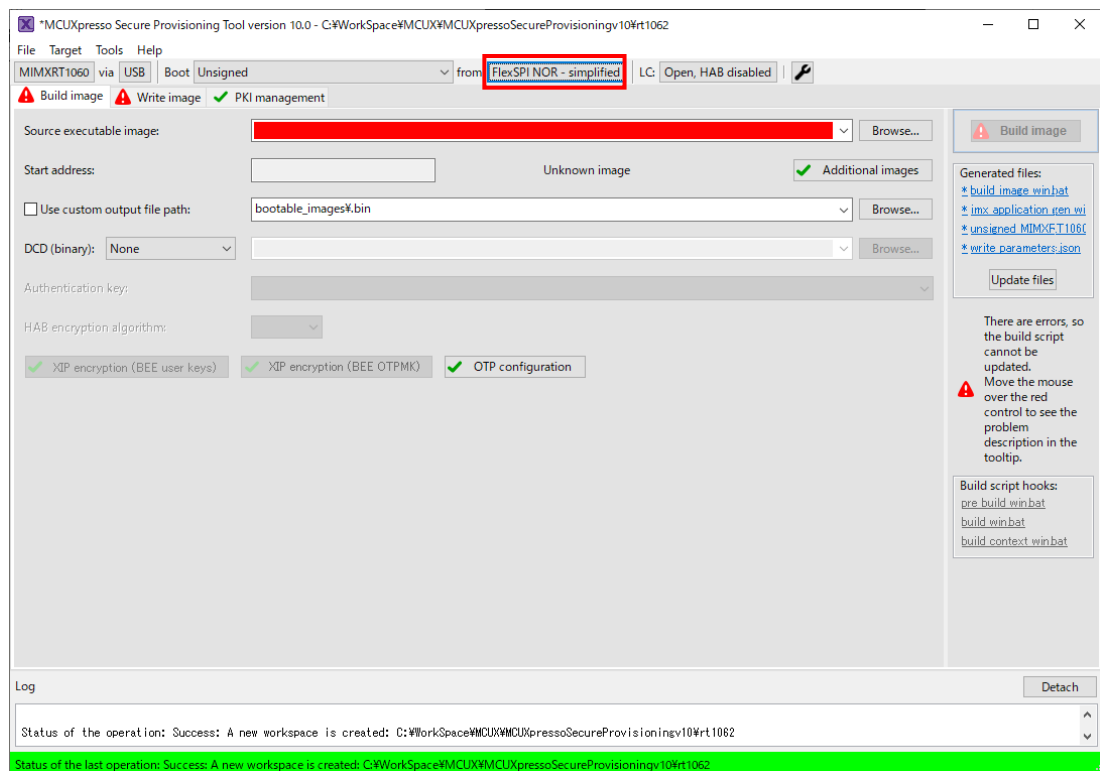


| ボード       | 選択する CPU 名                  |
|-----------|-----------------------------|
| NX-RT1021 | MIMXRT1021xxxxx(MIMXRT1020) |
| NX-RT1062 | MIMXRT1062xxxxx(MIMXRT1060) |
| NX-N947   | MCXN947                     |

- ⑧ 画面右下にある [Create] をクリックし、[New Workspace] を閉じます。



- ⑨ QSPI Flash の設定をするために、[from] の右側をクリックします。

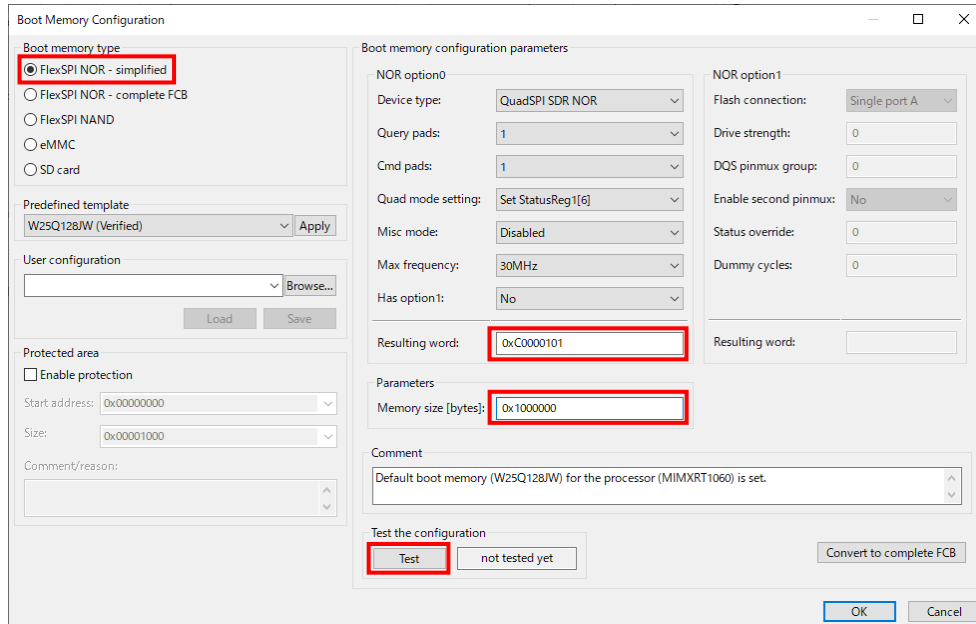


⑩ [Boot memory Configuration]が開きますので [Boot memory type] に [FlexSPI - simplified] を選択し、下記の処理を行います。

[Resulting word] に「0xC0000101」を設定します。

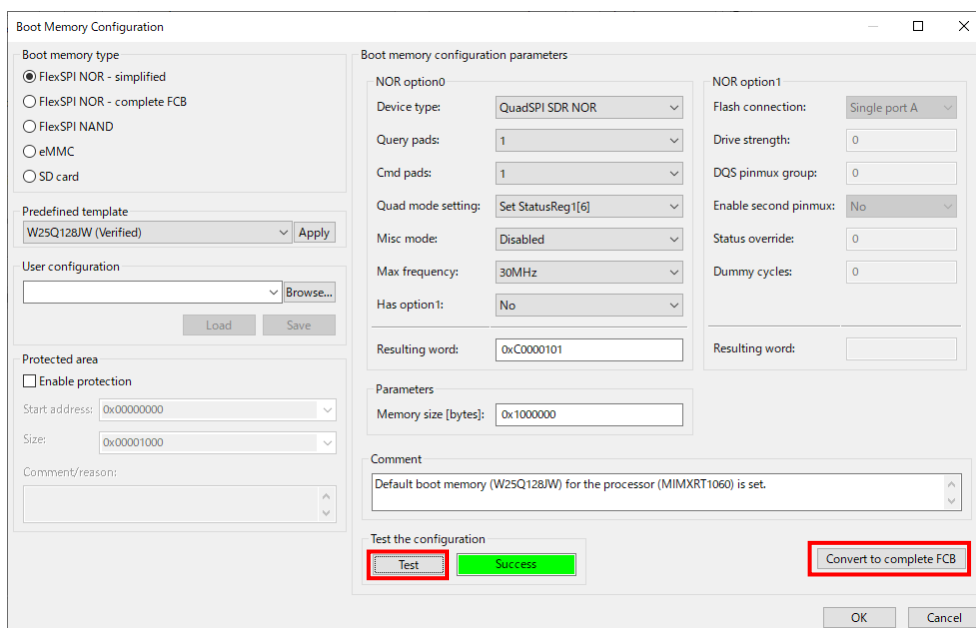
[Memory size] に対象のボードの QSPI Flash のサイズを設定します。（下記の表を確認してください。）

設定後、[Test the configuration] の [Test] をクリックします。

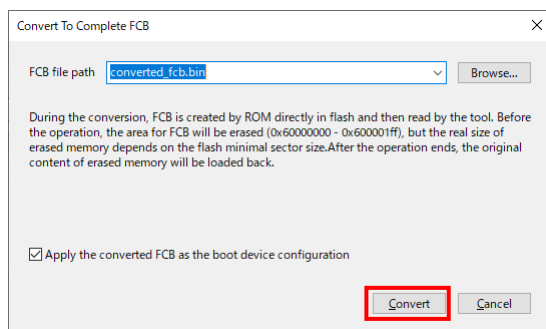


| ボード       | [Memory size]の設定値 |
|-----------|-------------------|
| NX-RT1021 | 0x400000          |
| NX-RT1062 | 0x1000000         |
| NX-N947   | 0x400000          |

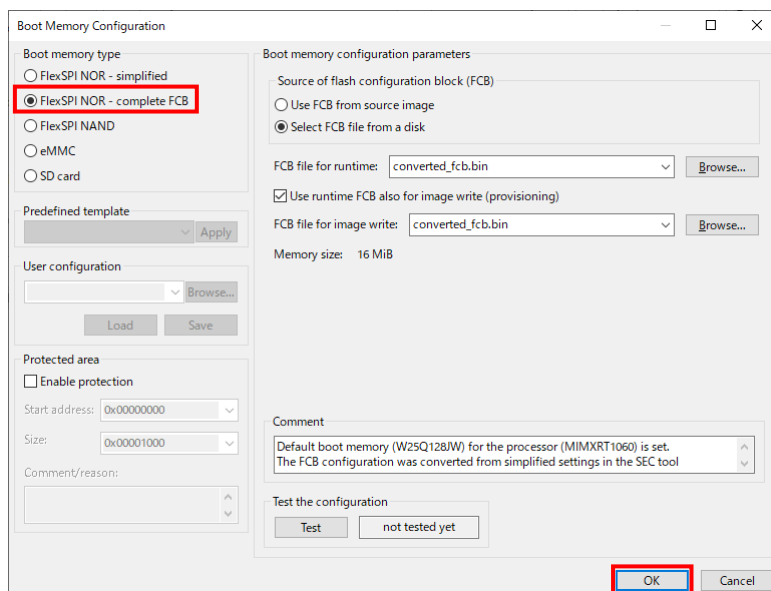
⑪ [Test the configuration] に [Success] と表示されることを確認し、[Convert to complete FCB]をクリックします。



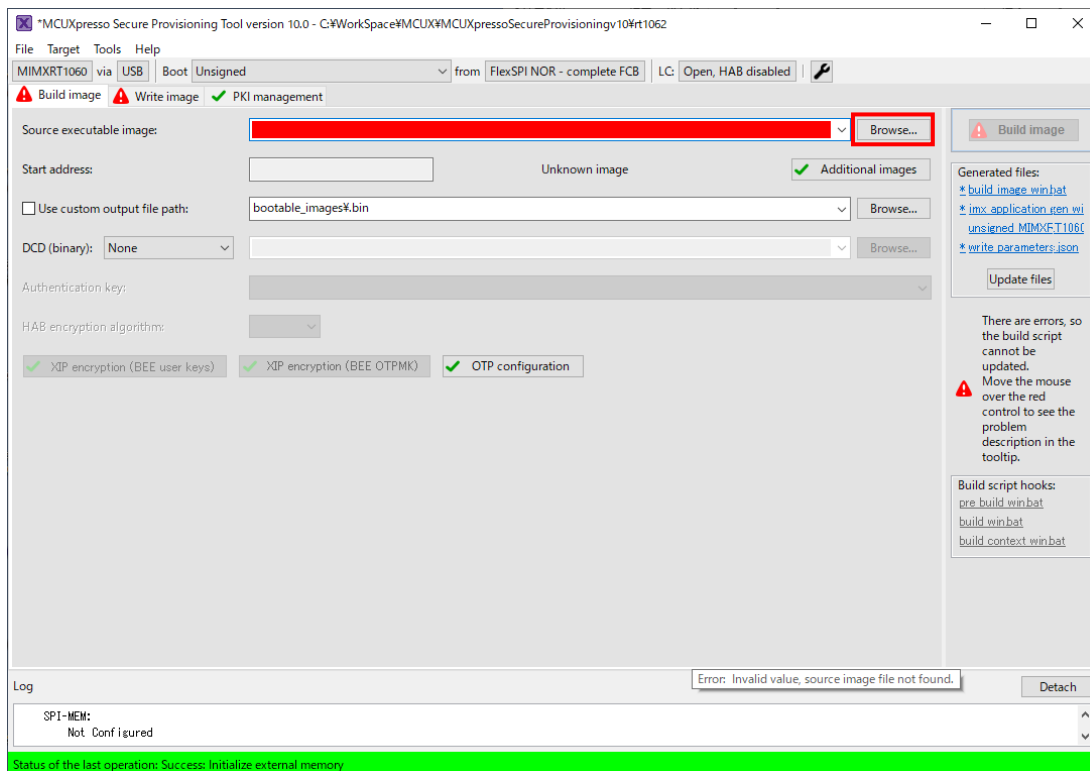
- ⑫ [Convert to complete FCB] が開くので、[Convert] をクリックします。



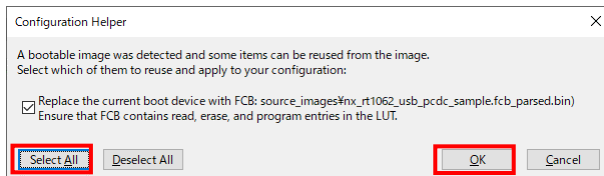
- ⑬ [Boot memory Configuration] の [Boot memory type] が [FlexSPI - complete FCB] 変化したことを確認し、[OK] をクリックします。



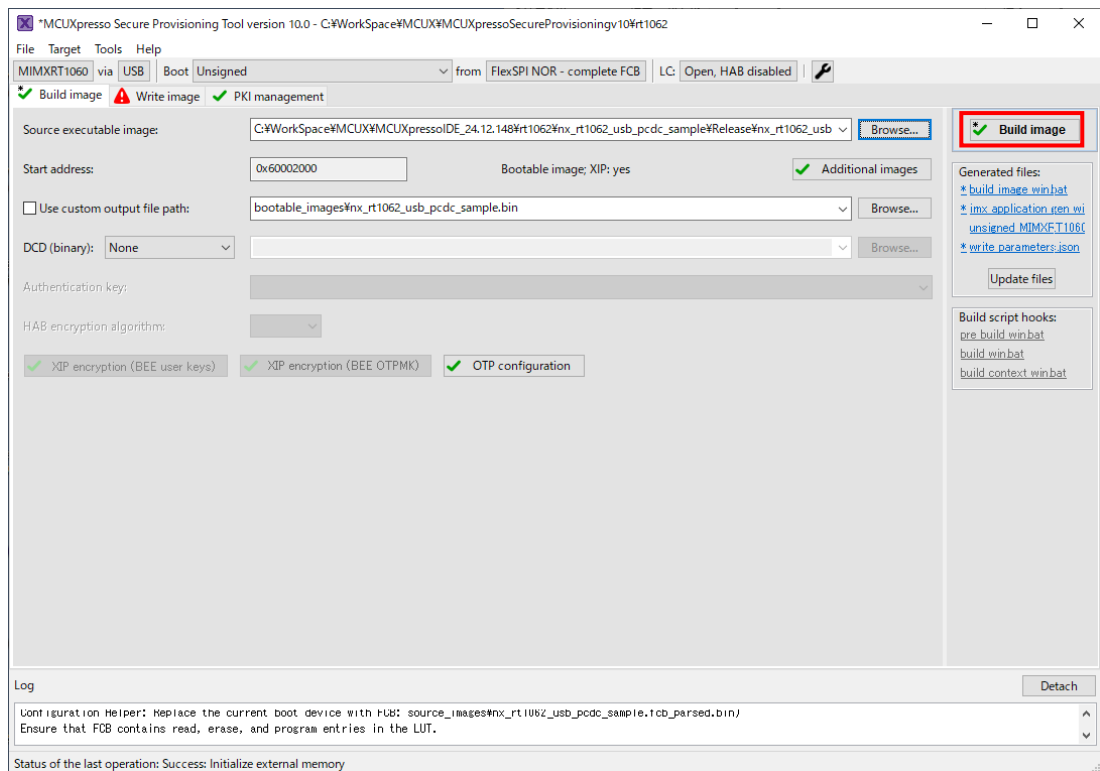
- ⑭ [Source executable image] の [Browse...] をクリックし、書き込みを行うファイル (\*.axf) を選択します。サンプルプログラムを書き込む場合には、ビルド構成が Release のものを選択してください。



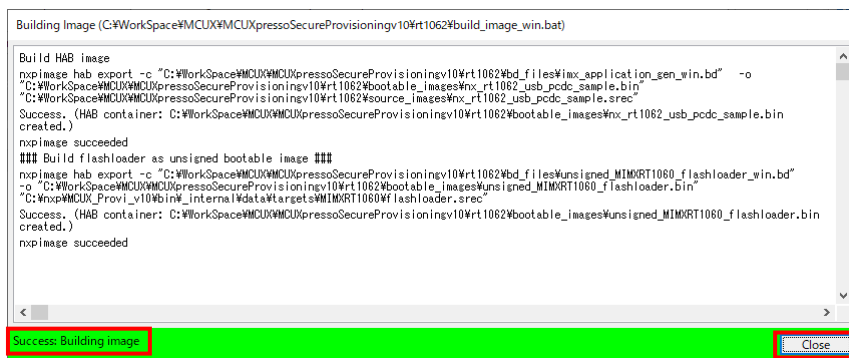
- ⑮ [Configuration Helper] が開いた場合、[Select All] をクリックした後、[OK] をクリックしてください。



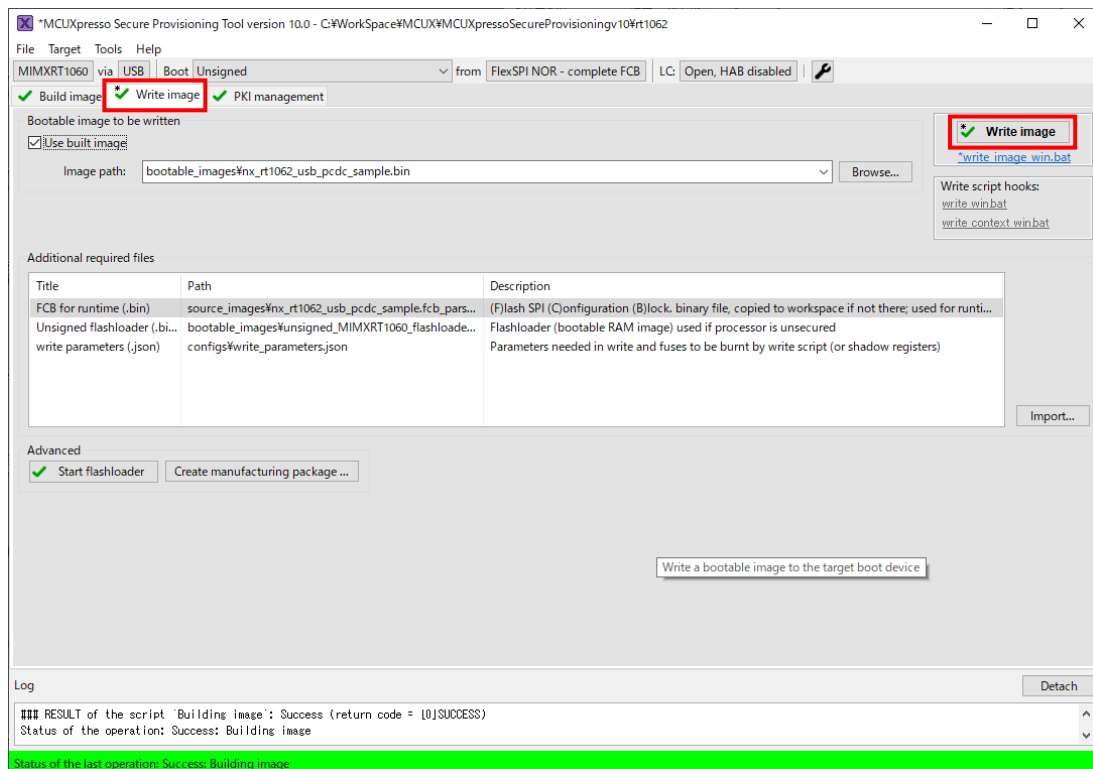
- ⑩ [Build image]をクリックします。



- ⑪ [Building image] が開き、イメージの作成が実行されます。  
イメージの作成に成功すると [Success: Building image] と表示されますので、[Close] をクリックします。



- ⑱ [Write image]タブをクリックし、表示された画面の [Write image] をクリックします。



- ⑲ [Write image] が開き、書き込み処理が行われます。  
書き込みが成功尾すると [Success:Writing image] と表示されますので、[Close] をクリックします。



- ⑳ 以上で、ボードへの書き込みは終了です。  
ボードの USB ケーブルを抜き、電源を切ります。

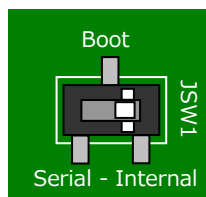
\*NX-RT1062 をお使いのお客様へ i.MX RT1060 エラッタ ERR010661 について

2ch の USB ポートが両方共“OTG”に設定されている場合、Vbus リークが発生します。  
本基板のサンプルプログラム等では、“USB Device Mode”レジスタ bit[1:0](リファレンスマニュアル 42.7.33 項)を、ch1 をデバイス、ch2 をホストに設定することにより、本エラッタによる弊害を回避しています。

シリアルブート時(JSW1 を“Serial Boot”に設定)は、USB ポートが両方共“OTG”に設定されてしまうため、本基板の電源を OFF する前に Type-C 側の USB ケーブルを取り外してください。Vbus リークによって本基板の電源が OFF しません。

- ② CPU ボードの設定を Internal Boot モード、Master Boot モードに変更し、書き込んだプログラムが動作することを確認します。

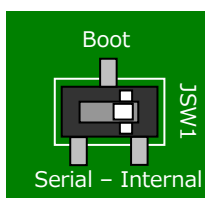
## &lt;NX-RT1021&gt;



JSW1 : Internal

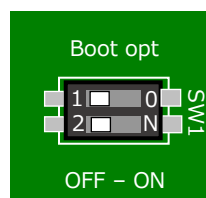
Internal Boot モード

## &lt;NX-RT1062&gt;



JSW1 : Internal

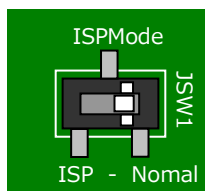
Internal Boot モード



SW1.1,1.2 : [OFF,OFF]

QSPI Flash Boot, Encrypted を使用しない

## &lt;NX-N947&gt;



JSW1 : Normal

Master Boot モード

再度同じ設定でファイルを書き込む際は、同じ Work space を開くことで②から作業を開始できます。

その他の機能など SEC ツールの詳細につきましては、MCUXpresso Secure Provisioning Tool User Guide を参照してください。

注意：Work space は、設定時のボードに対応しています。異なるボードへの書き込みには使用できません。

例えば、NX-RT1062 として作成した Work space を使用して、NX-RT1062 以外のボードに書き込むことはできませんので使用するボードに合わせた Work space を新規作成してください。

## 改定履歴

| 版数  | 更新日        | 改定内容 |
|-----|------------|------|
| 1 版 | 2025/05/16 | 新規作成 |

## ご注意

- ・本文書の著作権は株式会社アルファプロジェクトが保有します。
- ・本文書の内容を無断で転載することは一切禁止します。
- ・本文書に記載されているサンプルプログラムの著作権は株式会社アルファプロジェクトが保有します。
- ・本サンプルプログラムで使用されているミドルウェアおよびドライバの著作権は NXP セミコンダクターズ社が保有します。
- ・本文書に記載されている内容およびサンプルプログラムについてのサポートは一切受け付けておりません。
- ・本文書の内容およびサンプルプログラムに基づき、アプリケーションを運用した結果、万一損害が発生しても、弊社では一切責任を負いませんのでご了承ください。
- ・本文書の内容については、万全を期して作成いたしました。万一ご不審な点、誤りなどお気づきの点がありましたら弊社までご連絡ください。
- ・本文書の内容は、将来予告なしに変更されることがあります。

## 商標について

- ・ i.MX RT1062 は、NXP セミコンダクターズ社の登録商標、商標または商品名称です。
  - ・ i.MX RT1021 は、NXP セミコンダクターズ社の登録商標、商標または商品名称です。
  - ・ MCX N947 は、NXP セミコンダクターズ社の登録商標、商標または商品名称です。
  - ・ MCUXpresso IDE は、NXP セミコンダクターズ社の登録商標、商標または商品名称です。
  - ・ MCUXpresso SDK は、NXP セミコンダクターズ社の登録商標、商標または商品名称です。
  - ・ MCUXpresso Secure Provisioning は、NXP セミコンダクターズ社の登録商標、商標または商品名称です。
- 
- ・ Windows®の正式名称は Microsoft®Windows®Operating System です。
  - ・ Microsoft、Windows は、米国 Microsoft Corporation.の米国およびその他の国における商標または登録商標です。
  - ・ Windows®10、Windows®11 は、米国 Microsoft Corporation.の商品名称です。  
本文書では下記のように省略して記載している場合がございます。ご了承ください。  
Windows®10 は Windows 10 もしくは Win10  
Windows®11 は Windows 11 もしくは Win11
- 
- ・ その他の会社名、製品名は、各社の登録商標または商標です。



株式会社アルファプロジェクト

〒431-3114

静岡県浜松市中央区積志町 8 3 4

<https://www.apnet.co.jp>

E-MAIL : [query@apnet.co.jp](mailto:query@apnet.co.jp)