

emWin for RX QEスタートアップ・設定ガイド

Using “QE for Display [RX]”
RX72N Envision 評価ボード用

更新日：2023年10月



Embedded Software / Hardware Solutions

試作から量産ツールまで

「emWin for RX」について

RXグループMCUに対応するSEGGER社製「**emWin for RX**」GUIソフトウェアは、**RXマイコングループ（RXv1、RXv2、RXv3）** システム環境で無償で商用利用し、開発することが可能です。1999年から販売しており、emWinはワールドワイドで産業制御、スマートホーム・省エネ対応IoT機器、テレコム、自動車産業、家電、医療器械等、航空電子機器様々なアプリケーションで使用されています。emWin GUIはRXマイコンデバイスの内蔵RAMだけで使用出来ますので、外付けRAMが不要です。emWin GUIは使用中のOS（RX用embOS、μITRON等）で対応可、OS無しも使用可能です。**ロイヤリティー費用、量産制限等はありません。**

「emWin for RX」導入方法

SEGGER社WebPageからダウンロードします
[emWin for RX パッケージ（要ユーザー登録）](#)

https://www.segger.com/downloads/emwin/emWin_RX65N_RX72N_Envision_Kit.zip

[RX65N / RX72N Envision Kit用
emWinスタートアップガイド（PDF）](#)

https://www.embitek.co.jp/technote/emwin/StartupGuide_emWin_EnvisionKit.pdf

“QE for Display [RX]”をインストールして
Smart ConfiguratorからemWin
プロジェクトを作成します。
この資料内で手順を解説します。

参考資料：RXファミリ QE for Display [RX]
アプリケーションノート
rx-family-qe-display-rx-application-note

実行手順

1

事前準備

開発環境 e2Studio、
QE for Display [RX] を
インストールします

本セッションでは以下の開発環境を使用します。

- Host PC: Windows10 / Windows11
- 統合開発環境 e2 studio 2023-10
- ディスプレイ対応開発支援ツールパッケージ
QE for Display V3.1.1

2

BSPソースコード作成

QE for Display [RX] のSmart
Configuratorを使用してemWin+LCD表示
に必要なソースコード及びプロジェクトを
作成します。

作成内容：

- ビルド用プロジェクト
- 基本BSP (CPU、クロック、メモリ設定)
- LCD、DMA、タッチパネル操作に必要なドライ
バファイル
- emWinモジュール (ライブラリ+共通ヘッダ)

3

画面表示の作成

【オプション1】
GUI描画ツールAppWizardから
画面をデザインしてemWinアプリ
ケーションコードを作成します。

----- 又は -----

【オプション2】
ユーザ指定のemWinアプリケー
ションソースを使用します。
※ AppWizardなしでウィジェット又は
GUI APIのemWinアプリケーションも使
用可能です。

TASK-1

事前準備

開発環境 e2Studio、QE for Display [RX] を
インストールします

本セッションでは以下の開発環境を使用します。

- Host PC: Windows10 / Windows11
- 統合開発環境 e2 studio 2023-10
- ディスプレイ対応開発支援ツール パッケージ QE for Display V3.1.1

以下のWebリンクから「統合開発環境e²studio 2023-10 インストーラ」プログラムをダウンロードしてインストールします。

https://www.renesas.com/e2studio_download

統合開発環境 e² studio

概要

ダウンロード

ドキュメント

設計・開発

調べる

サポート

ビデオ&トレーニング

詳細情報

↓ ダウンロード

Start typing to filter results by title



全種類

日付

統合開発環境 e² studio 2023-10 Linux用

🔒 RUN 1,363.14 MB 英語

アップデート-統合開発環境

2023年10月20日

統合開発環境e² studio 2023-10 Windows用インストーラ

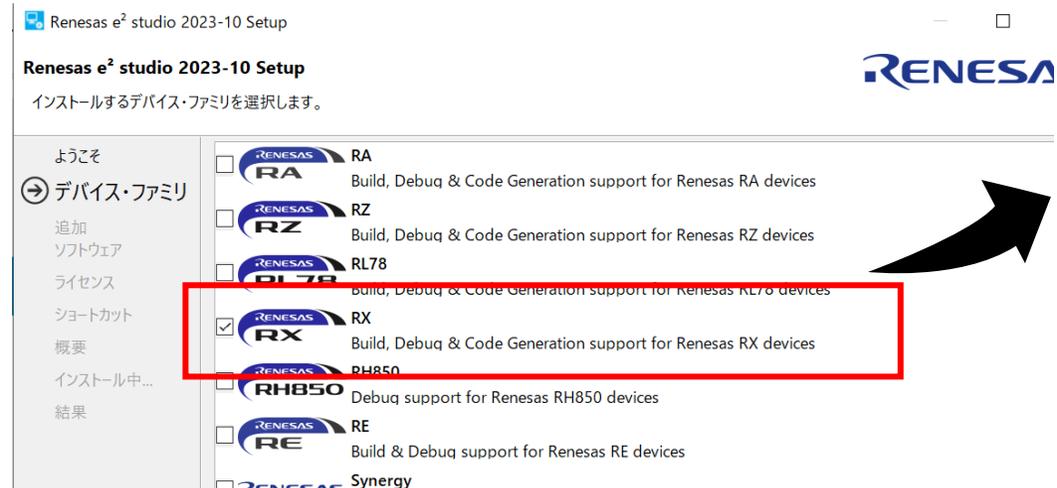
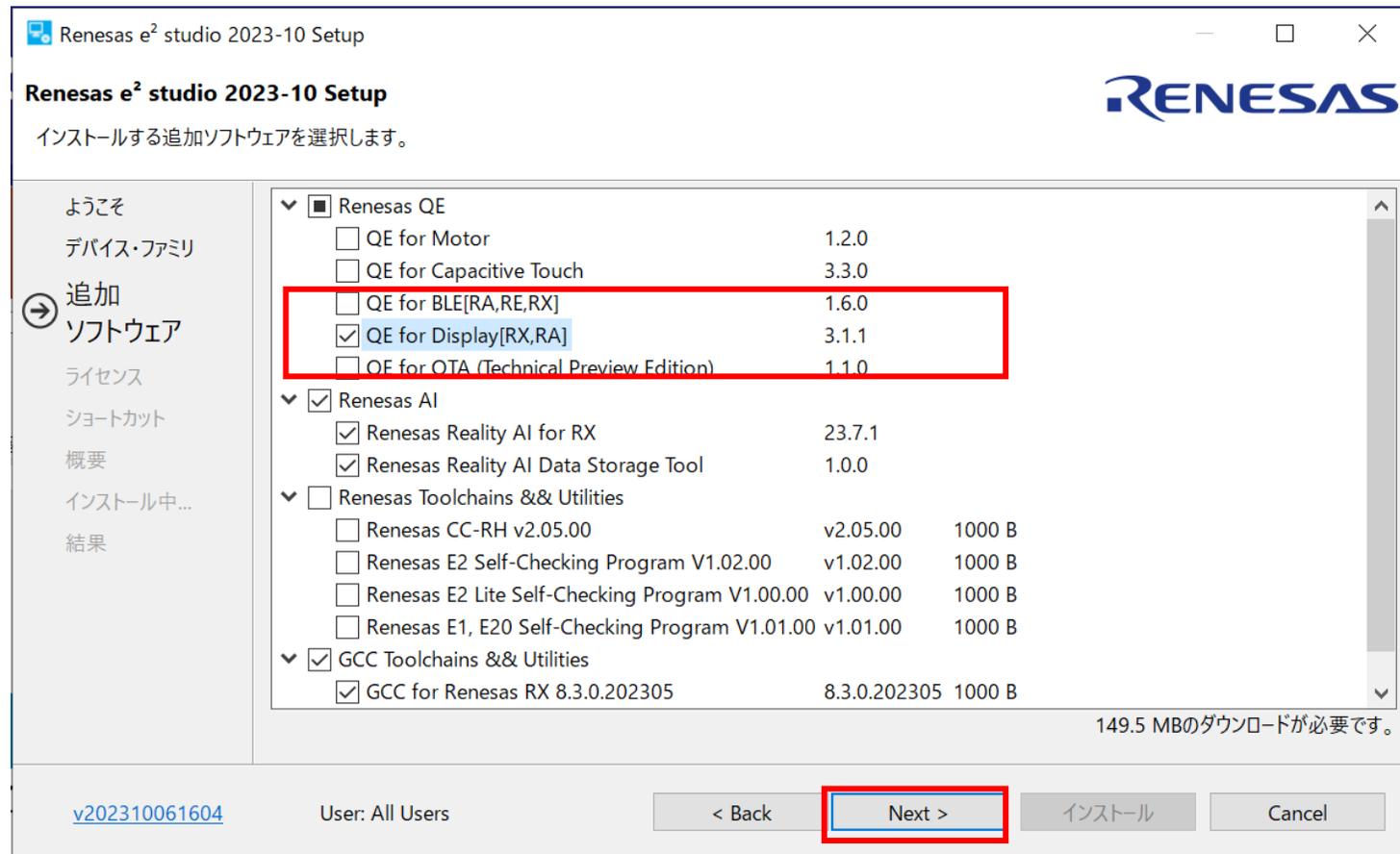
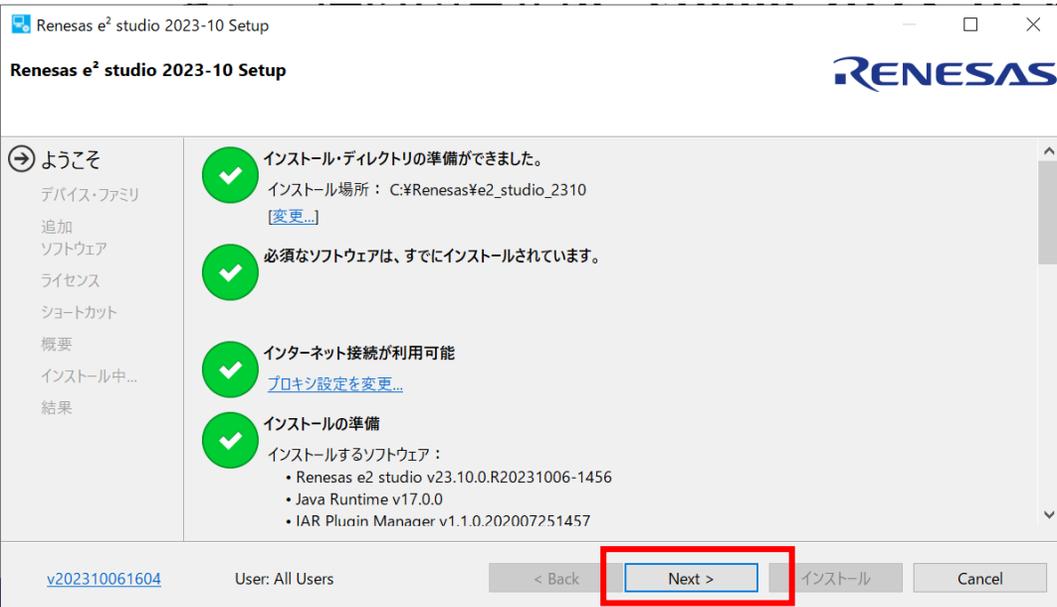
🔒 ZIP 1,444.24 MB 英語

アップデート-統合開発環境

2023年10月20日

事前準備：開発環境のインストール

インストールの「追加ソフトウェア」で、**QE for Display** を選択してください。



インストール済みe²studio から「QE for Display」をダウンロード→インストールは QE for Display [RX] アプリケーションノートの手順をご参照ください。
「rx-family-qe-display-rx-application-note」

RX72N Envision Kit



マイコン：RX72N MCU（RXv3コア 240 MHz動作）

内蔵ROM：コードフラッシュメモリ：4MB

内蔵RAM：1MB

デバッガI/F：E2 Liteエミュレータ

静電容量性タッチ機能付き4.3インチWQVGA TFTパネル

RX72N Envision Kitの詳細はこちら

<https://www.renesas.com/jp/ja/products/software-tools/boards-and-kits/eval-kits/rx72n-envision-kit.html>

TASK-2

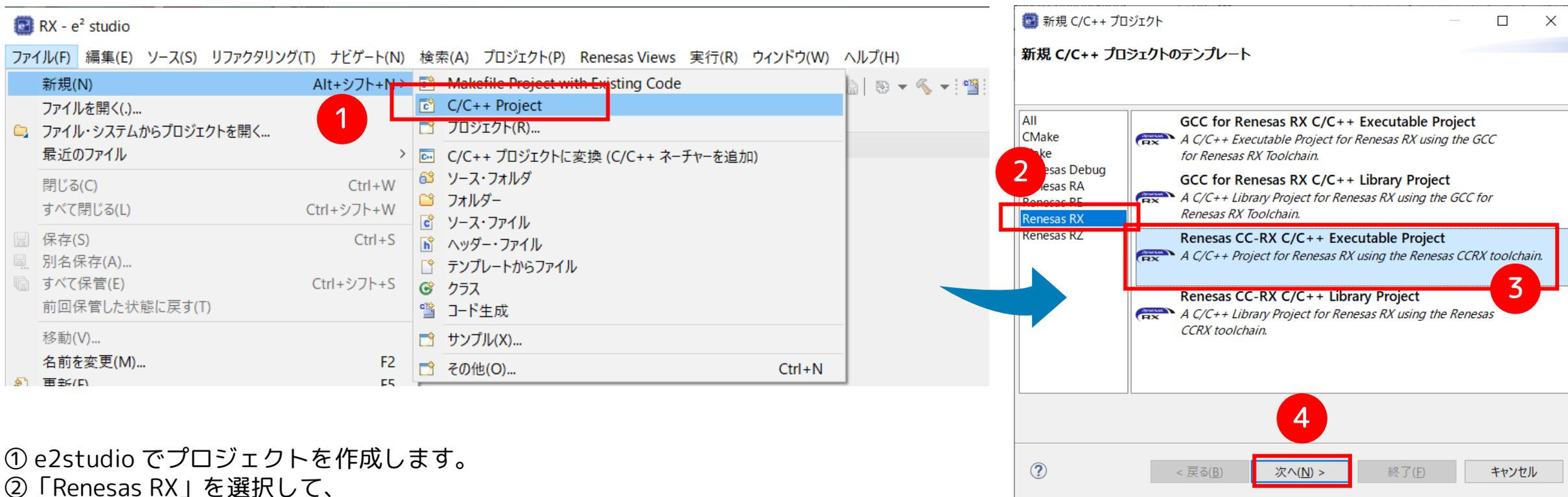
RX72N Envisionボード用 BSPソースコード作成

QE for Display [RX] のSmart Configuratorを使用して
emWin+LCD表示に必要なソースコード及び
プロジェクトを作成します。

実行手順：

- ・ Envisionボード用プロジェクト作成
- ・ クロック設定
- ・ HEAPメモリ設定
- ・ SW2押しボタンポート設定
- ・ LCDインターフェース設定
- ・ LCDコントローラの導入
- ・ LCDの表示調整
 - ・ TCON/LCD設定
 - ・ 表示タイミング調整
 - ・ 画質調整
- ・ ファイル出力
- ・ 画面表示の作成
 - ・ emWinを導入
 - ・ emWinの初期設定
 - ・ ファイル出力
- ・ プロジェクトビルド確認
- ・ デバッグ設定

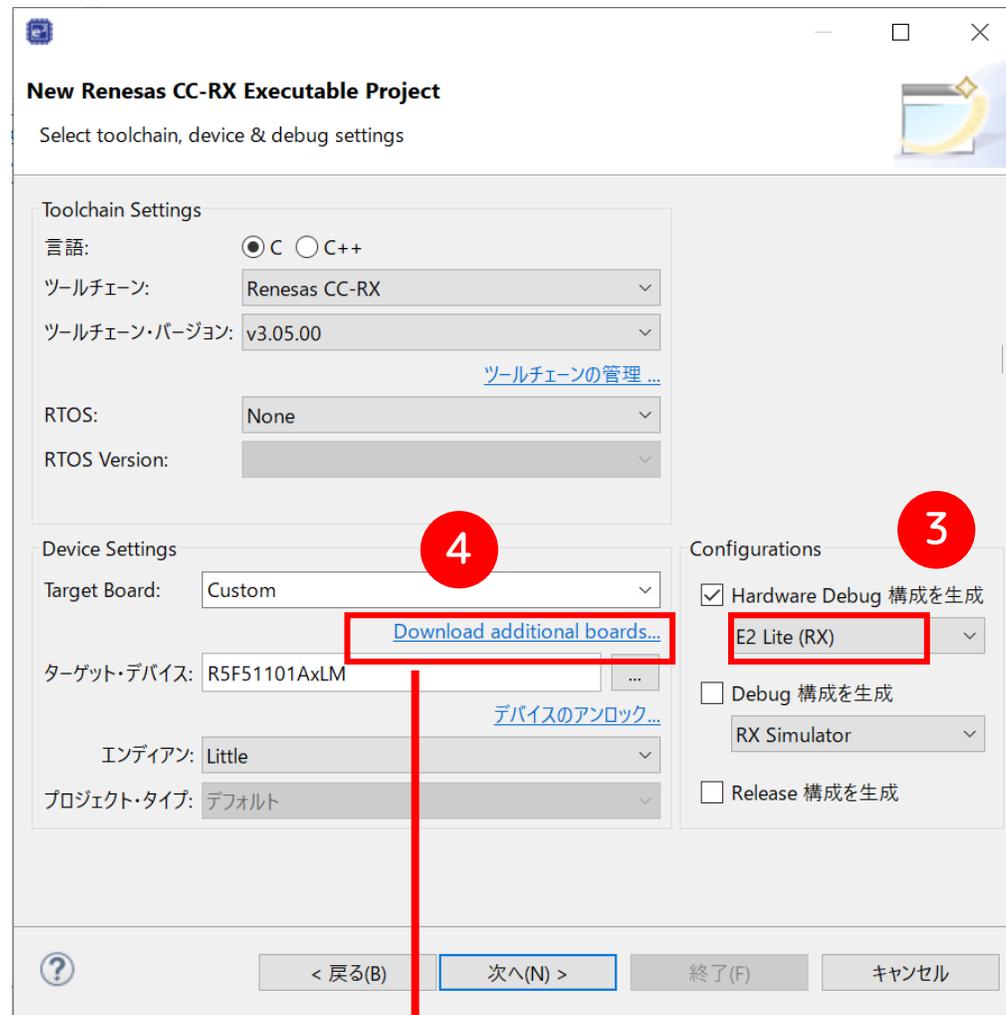
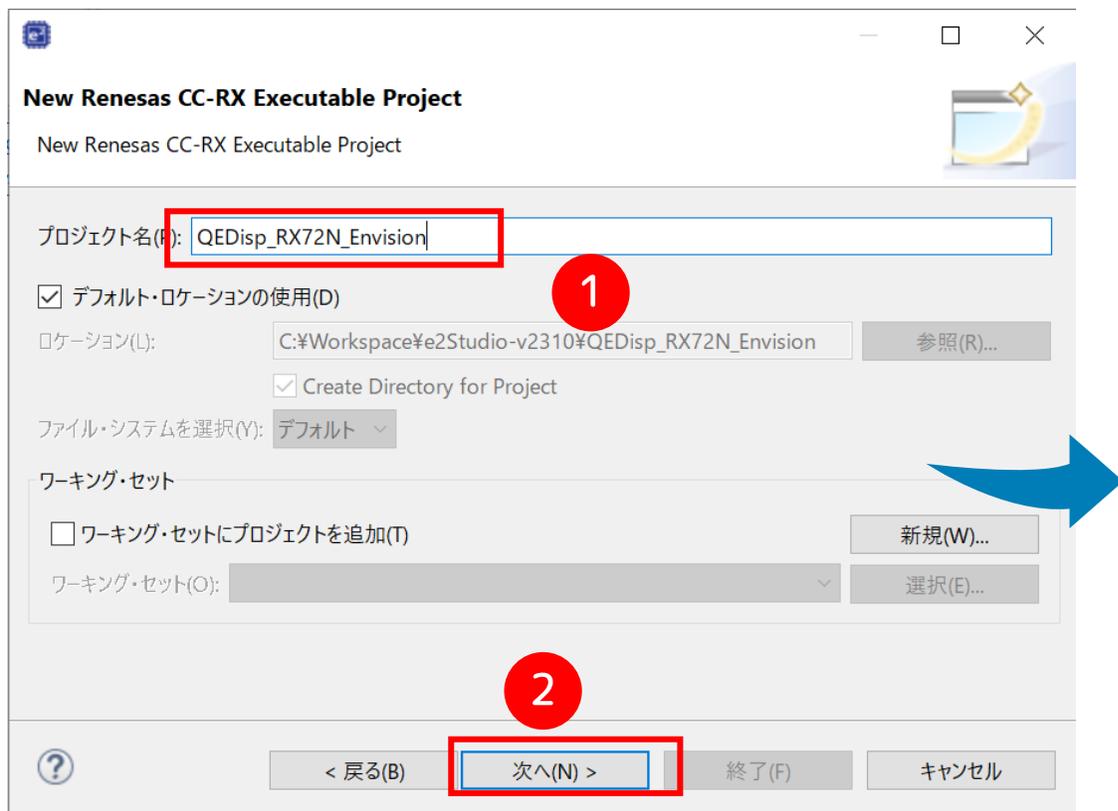
QE for Display [RX] のSmart Configuratorを使用してemWin+LCD表示に必要なソースコード及びプロジェクトを作成します。



- ① e2studio でプロジェクトを作成します。
- ② 「Renesas RX」を選択して、
- ③ CCRXコンパイラ用「Renesas CC-RX C/C++ Executable Project」を選択します。
- ④ 「次へ(N)>」をクリックします。

プロジェクト作成：プロジェクト名、ボード名設定

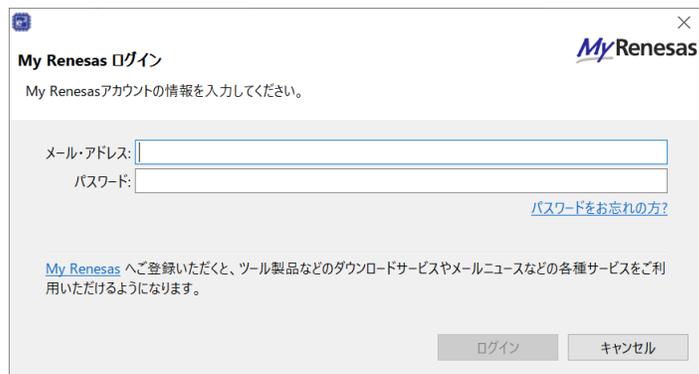
①[プロジェクト名(P):]のテキストボックスにプロジェクト名を入力し、②[次へ(N)>]をクリックします。
本セッションでは、プロジェクト名に「QEDisp_RX72N_Envision」を使用します。



エミュレータに「E2 Lite (RX)」を選択します。

[Target Board:]の選択肢に「EnvisionKitRX72N」ボードがない場合は、Board Description File (BDF) のダウンロードが必要です。
その場合、「Download additional boards..」をクリックしてください。

プロジェクト作成：プロジェクト名、ボード名設定



My Renesas ログイン

My Renesasアカウントの情報を入力してください。

メール・アドレス:

パスワード: [パスワードをお忘れの方?](#)

My Renesas へご登録いただくと、ツール製品などのダウンロードサービスやメールニュースなどの各種サービスをご利用いただけるようになります。

ログイン設定後に、

①「**Envision Kit for RX72N** …」ボードを選択して

②[ダウンロード]をクリックします。



ボード定義ファイルのダウンロード

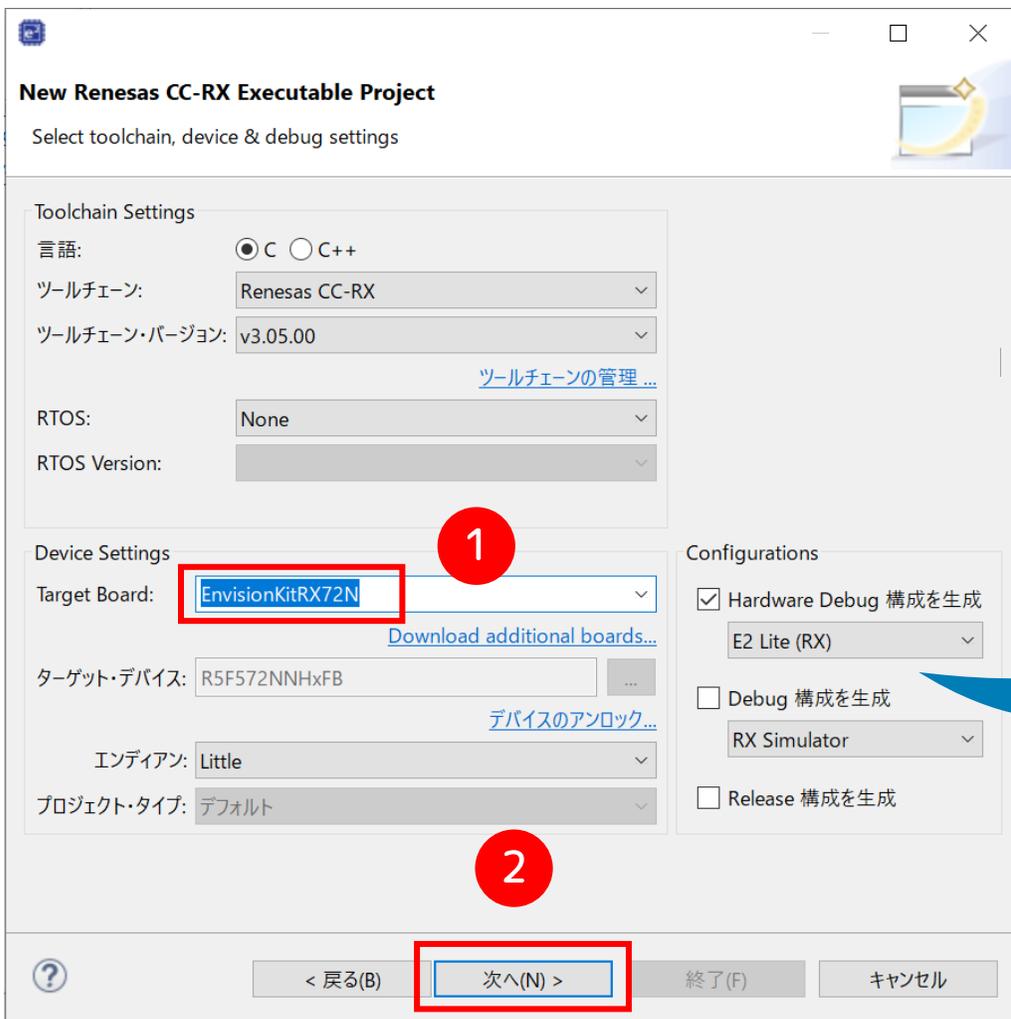
ダウンロードするボード定義ファイルを選択してください。

| タイトル | リビジョン |
|--|-------------|
| <input type="checkbox"/> Envision Kit for RX65N Board Description File | 1.14 |
| <input type="checkbox"/> Cloud Kit for RX65N Board Description File | 1.13 |
| <input checked="" type="checkbox"/> Envision Kit for RX72N Board Description File | 1.13 |
| <input type="checkbox"/> Target Board for RX130 Board Description File | 1.12 |
| <input type="checkbox"/> Target board for RX231 Board Description File | 1.12 |
| <input type="checkbox"/> Target board for RX23W Board Description File | 1.12 |
| <input type="checkbox"/> Target Board for RX65N Board Description File | 1.12 |
| <input type="checkbox"/> Renesas Starter Kit+ for RX65N-2MB Board Description File | 1.05 |
| <input type="checkbox"/> Renesas Starter Kit for RX66T Board Description File | 1.04 |
| <input type="checkbox"/> CPU Card for MCU Evaluation RX66T Board Description File | 1.03 |

モジュール・フォルダー・パス:

プロジェクト作成：ボード名、Configurator設定

- ① [Target Board:] に「EnvisionKitRX72N」ボードを選択します。
- ② [次へ(N)>]をクリックします。



- ③ [Use Smart Configurator]のチェックボックスにチェックを入れます。
- ④ [終了(F)]をクリックします。

プロジェクト作成：スマート・コンフィグレータを開きます

The screenshot displays the e2Studio IDE interface. The main window shows the 'Smart Configurator' overview page, which includes sections for '概要' (Overview), '動画' (Videos), '最新情報' (Latest Information), and '製品ドキュメント' (Product Documents). A diagram on the right illustrates the software stack: 'Application Code' at the top, followed by 'Software Components' (containing 'RTOS' and 'Middleware & Drivers'), and 'Device Drivers' at the bottom, all supported by 'MCU Hardware'. A 'Smart Configurator' tool is shown interacting with the software components. A dialog box in the foreground asks '関連付けられたパースペクティブを開きますか?' (Do you want to open the associated perspective?). The 'はい(Yes)' button is highlighted with a red box. The console window at the bottom shows the output of the Smart Configurator, including file generation and code completion messages. A small notification in the bottom right corner indicates that the Smart Manual is being installed.

関連付けられたパースペクティブを開きますか?

スマート・コンフィグレータ パースペクティブを開きますか?

常にこの設定を使用する(R)

パースペクティブを開く(O) いいえ(N)

スマート・コンフィグレータ出力

```
M04000001: ファイルを生成: src\smc_gen\general\r_cg_userdefine.h
M04000001: ファイルを生成: src\smc_gen\general\r_smc_entry.h
M04000001: ファイルを生成: src\smc_gen\general\r_cg_hardware_setup.c
M05000012: ファイルを生成: src\smc_gen\r_pincfg\Pin.h
M05000012: ファイルを生成: src\smc_gen\r_pincfg\Pin.c
M06000002: ファイルを生成: src\smc_gen\general\r_smc_interrupt.c
M06000002: ファイルを生成: src\smc_gen\general\r_smc_interrupt.h
M06000002: ファイルを生成: src\smc_gen\r_config\r_bsp_interrupt_config.h
M03000004: ファイルを変更: src\smc_gen\r_config\r_bsp_config.h
M00000002: コード生成の終了: C:\Workspace\e2Studio-v2310\QEDisp_RX72N_Envision\src\smc_gen
```

コンフィグレーションチェック

| 記述/説明 | 型 |
|-------|---|
| 0 項目 | |
| 記述/説明 | |

スマート・マニュアルをインストール中...

新しいスマート・マニュアルがインストールされています。

スマート・マニュアルの変更 [インストール設定](#)

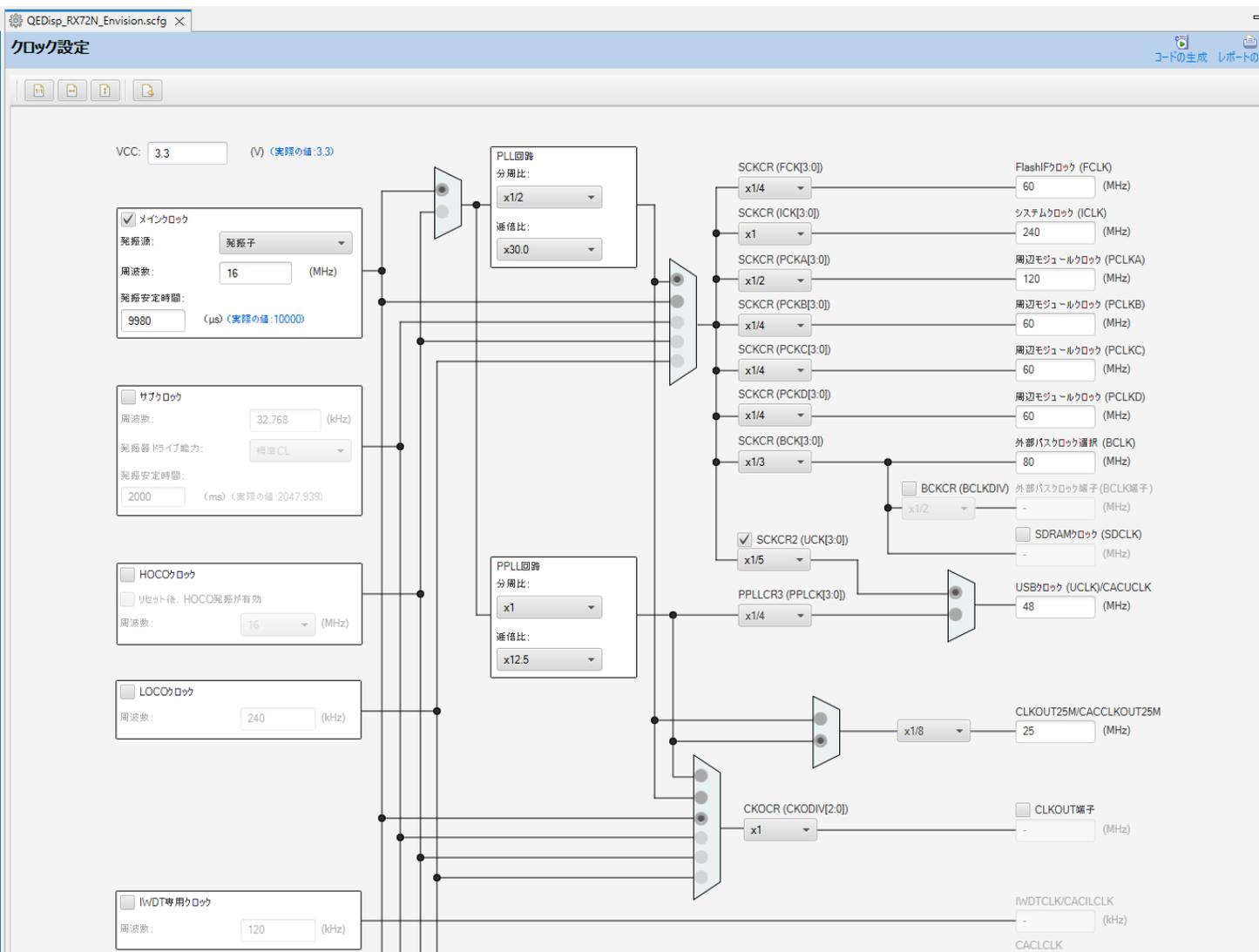
プロジェクト作成：スマート・コンフィグレータ設定（クロック）



- ① [ボード] グループを選択します。
- ② ボード名[EnvisionKitRX72N]を確認します。
- ③ [クロック]グループを選択します。
- ④ クロック設定を確認します



「EnvisionKitRX72N」ボードの場合は：メインクロック：16MHz、CPUクロック：240MHz



プロジェクト作成：スマート・コンフィグレータ設定（HEAPメモリ）

ソフトウェアコンポーネント設定

コンポーネント 設定

フィルタ入力

Startup

ジェネリック

r_bsp

| プロパティ | 値 |
|--|----------------------------|
| Configurations | |
| # User stack setting | 2 stacks |
| # User stack size | 0x1000 |
| # Interrupt stack size | 0x400 |
| # Heap size | 0x4000 |
| # Initializes C input and output library functions | Enable |
| # Enable user stdio charget function | Use BSP charget() function |
| # User stdio charget function name | my_sw_charget_function |
| # Enable user stdio charput function | Use BSP charput() function |
| # User stdio charput function name | my_sw_charput_function |
| # Processor Mode | Stay in Supervisor mode |
| # ID code 1 | 0xFFFFFFFF |
| # ID code 2 | 0xFFFFFFFF |
| # ID code 3 | 0xFFFFFFFF |

Macro definition: BSP_CFG_HEAP_BYTES
Heap size
NOTE: This setting is available only when using CCRX and GNUC.

概要 ボード クロック システム コンポーネント 端子 割り込み

- ① [コンポーネント] グループを選択します。
- ② [r_bsp]コンポーネントを選択します。
- ③ [Heap size] 項目のメモリサイズを **0x4000** に増やします。このメモリはRX DRW2Dエンジン用のメモリプールに使用されます。

プロジェクト作成：スマート・コンフィグレータ設定（SW2ポート）

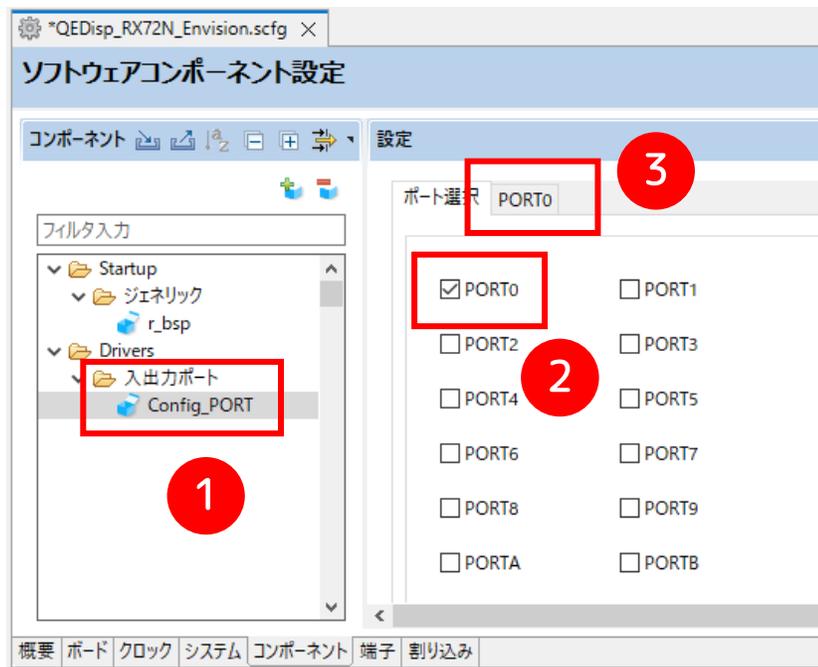
オンボードSW2プッシュボタンポート「P07」をGPIOモードに設定します。

The screenshot shows the 'ソフトウェアコンポーネント設定' (Software Component Settings) window. The 'コンポーネント' (Component) tab is active. The 'コンポーネント' list on the left shows 'r_bsp' selected. The '追加' (Add) icon is highlighted with a red box and a blue arrow pointing to the 'コンポーネントの追加' dialog. The dialog shows the 'ポート' (Port) component selected in the list. The 'Config_PORT' configuration name and 'PORT' resource are shown. The '終了(F)' (Finish) button is highlighted with a red box and a blue arrow.

① [コンポーネント] グループを選択します。
② [コンポーネントの追加] アイコンをクリックします。
③ ④ ⑤ [ポート] ソフトウェアドライバコンポーネントを追加します。

| コンポーネント | Short Name | タイプ |
|--------------------|------------|-------|
| バス | | コード生成 |
| プログラマブルパルスジェネレータ | | コード生成 |
| ポート | | コード生成 |
| ポートアウトプットインテグレーション | | コード生成 |

プロジェクト作成：スマート・コンフィグレータ設定（SW2ポート）



- ① [コンポーネント] グループの[Config_PORT]を選択します。
- ② ③ [ポート選択]タブ下の[PORT0]項目にチェックを入れます。
- ④ [PORT0]タブを選択します。
- ⑤ P07グループ設定を「GPIO入力ポート」モードに変更します。

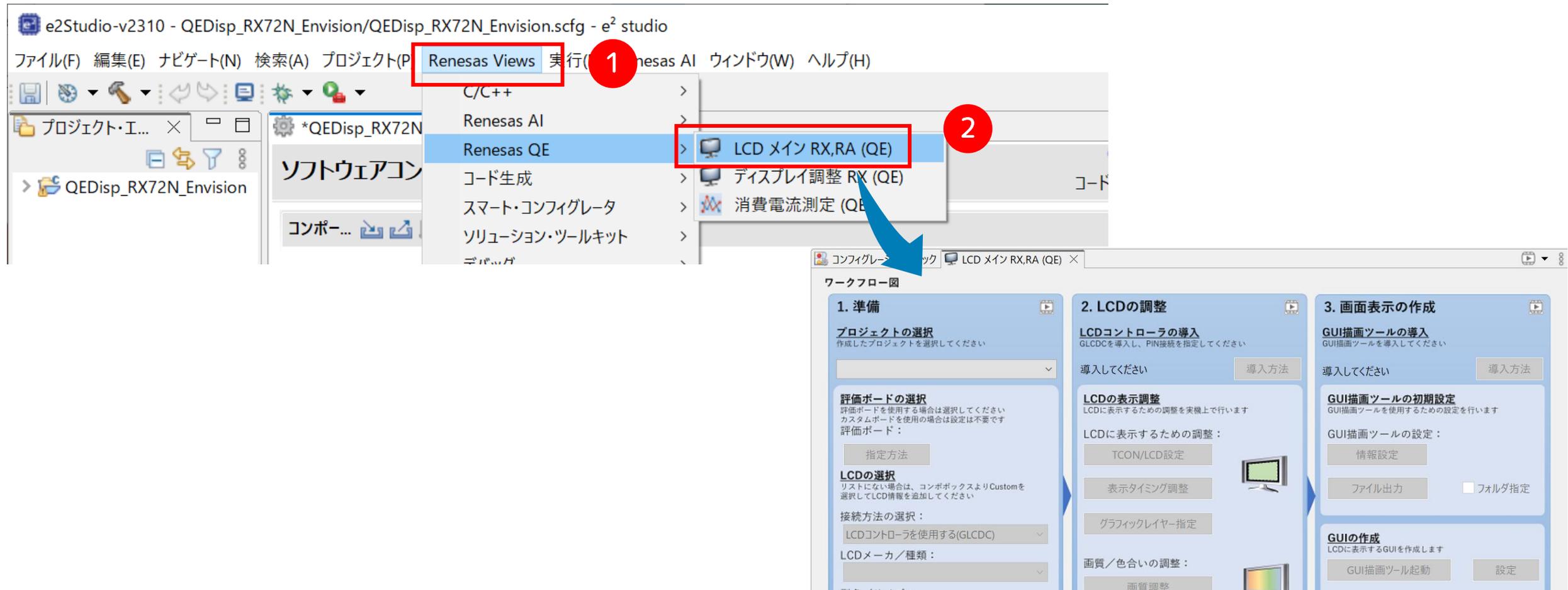
オンボードSW2押しボタンポート「P07」をGPIOモードに設定します。

プロジェクト作成：LCDインターフェース設定

QE for Display [RX]を起動してLCDインターフェース関連の
コンフィグレーション設定を行います

実行手順：

メニューから[Renesas Views]→[Renesas QE] →[LCD メインRX (QE)]コマンドを実行します。

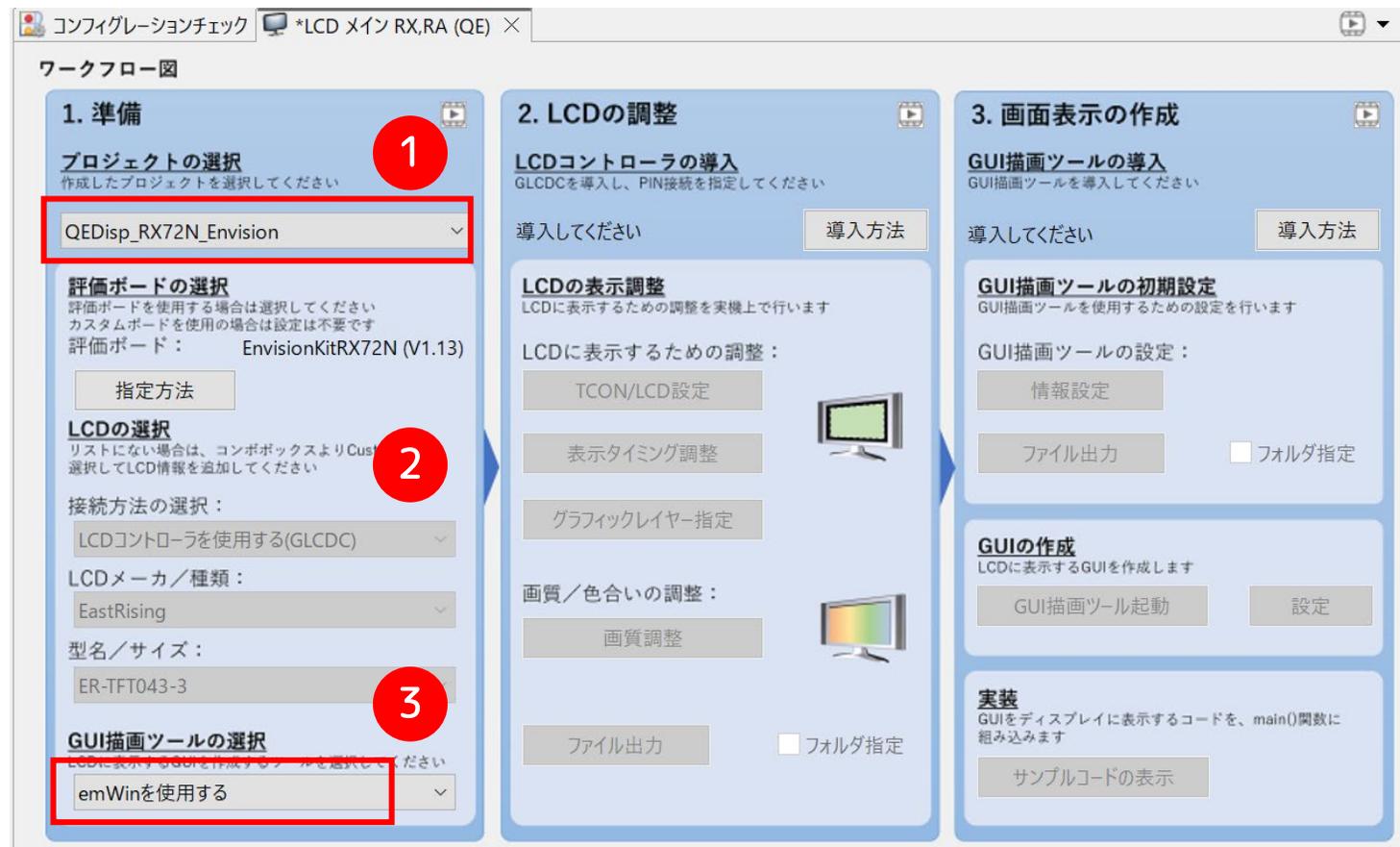


「LCD メインRX (QE)」ワークフローの設定を行います。

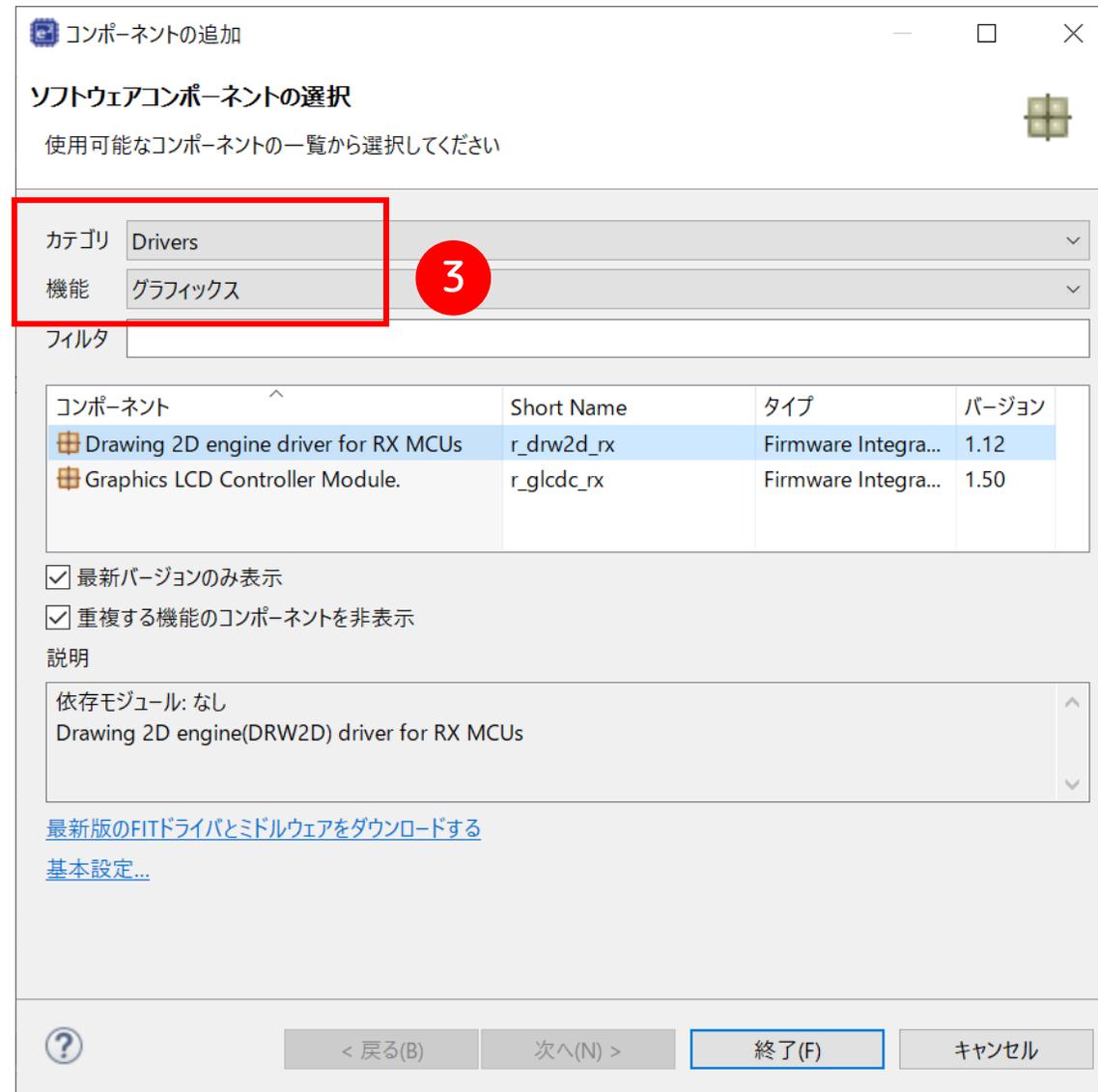
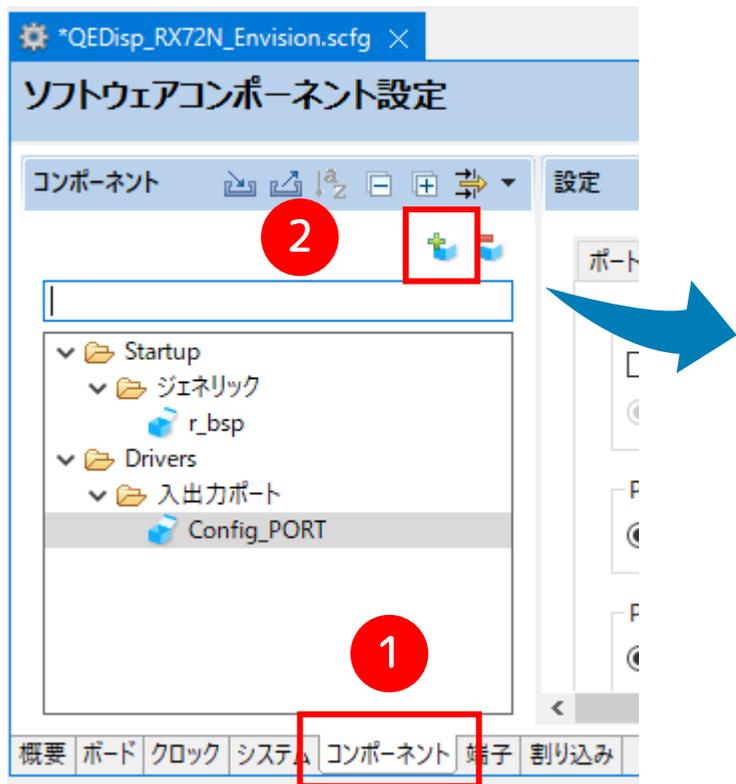
① [プロジェクトの選択]で、対象のプロジェクト「**QEDisp_RX72N_Envision**」を選択します。

② LCDメーカ/品名を確認します。
RX72N Envisionボードは以下の通りです。
LCDメーカ/種類：EastRising
型名/サイズ：ER-TFT043-3

③ [GUI 描画ツールの選択]に [**emWin を使用する**]を選択します。



プロジェクト作成：LCDコントローラの導入（ドライバモジュール）

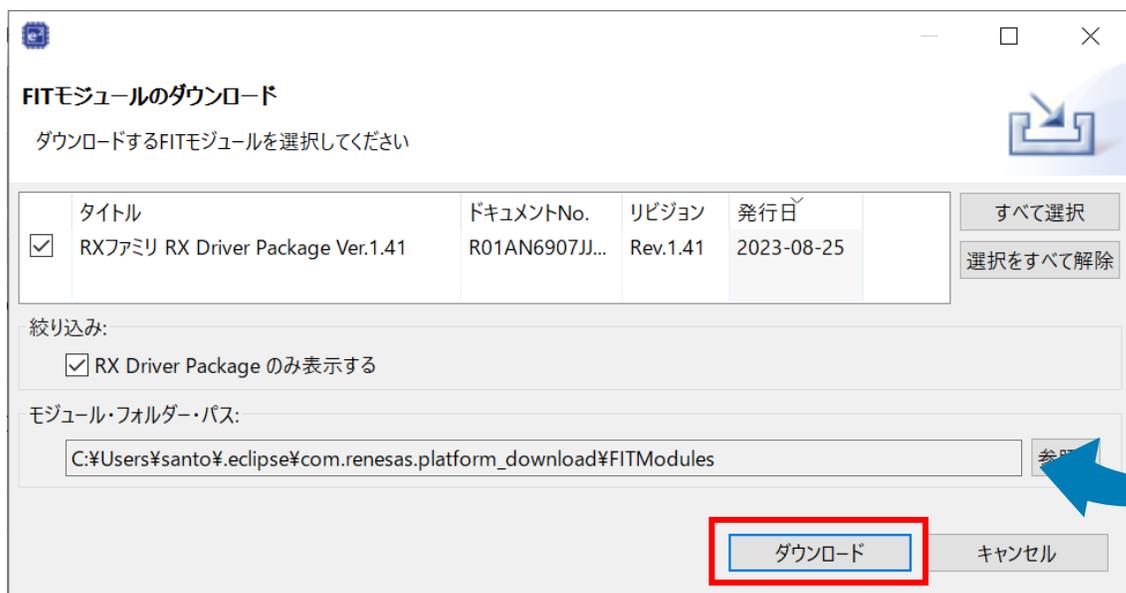


- ① [コンポーネント] グループを選択します。
- ② [コンポーネントの追加]アイコンをクリックします。
- ③ [カテゴリ：Drivers]そして[機能]項目に「グラフィックス」を選択してLCD関連のドライバモジュールを確認します。

プロジェクト作成：LCDコントローラの導入（ドライバモジュール）

④ 以下のドライバモジュールが**現れない場合は**、**[最新版のFITドライバとミドルウェアをダウンロードする]**をクリックして、最新版のドライバモジュールをインストールします。

- Drawing 2D engine(DRW2D) driver for RX MCUs [r_drw2d_rx]
- Graphics LCD Controller Module [r_glcd_rx]



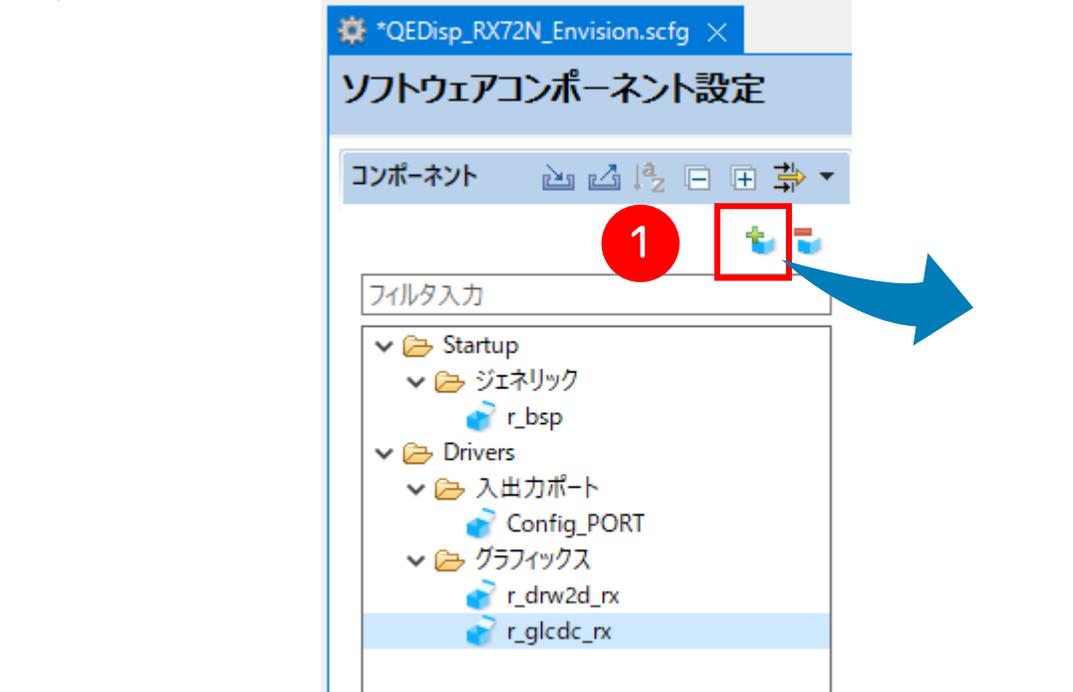
⑤ [r_drw2d_rx]、[r_glcd_rx]ドライバモジュールを選択して、

⑥ [終了(F)]ボタンをクリックします。



プロジェクト作成：LCDコントローラの導入（ドライバモジュール）

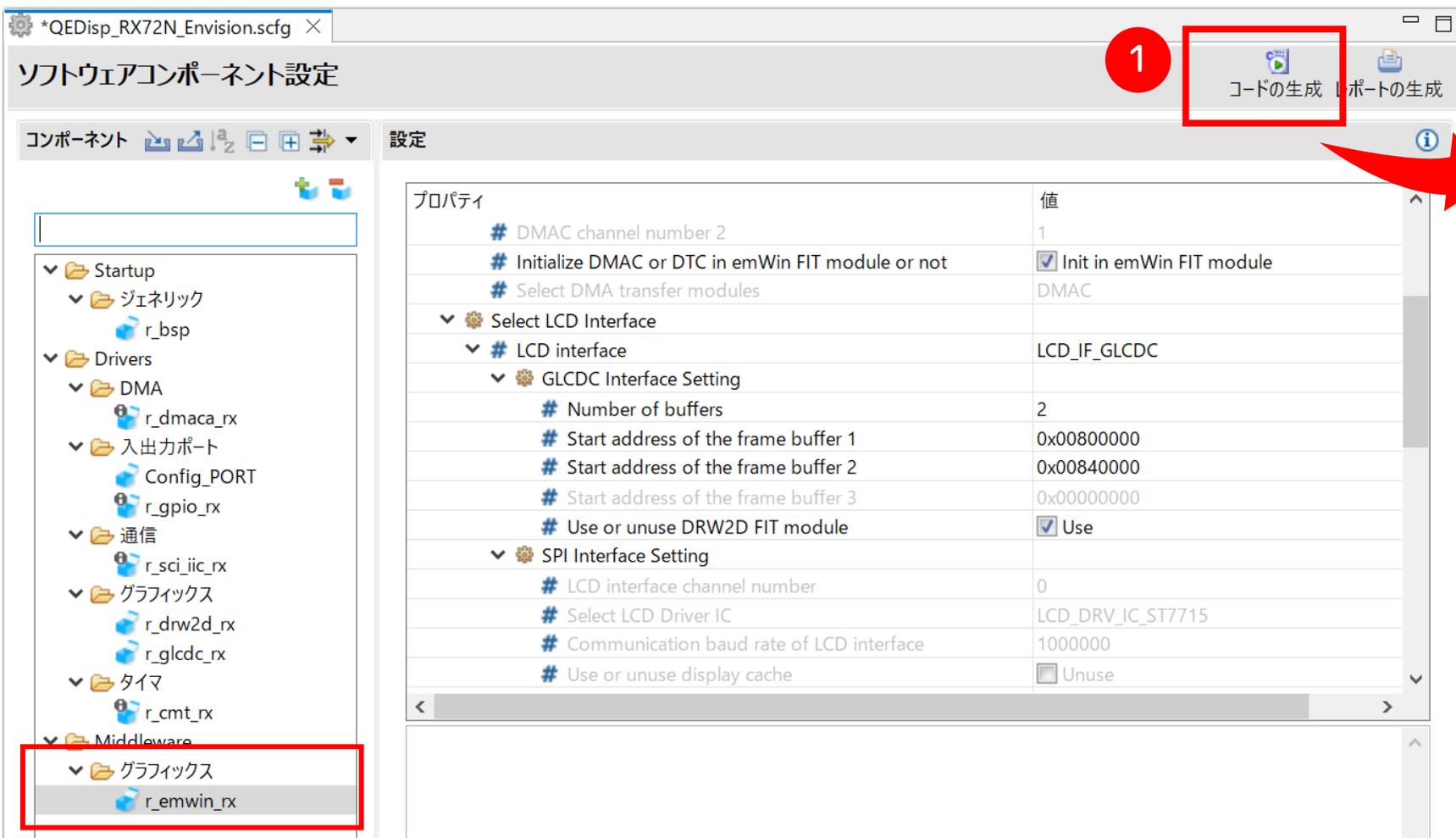
ドライバ追加されましたら、コンポーネントウィンドウに[r_drw2d_rx]、[r_glcd_rx] モジュールが表示されま



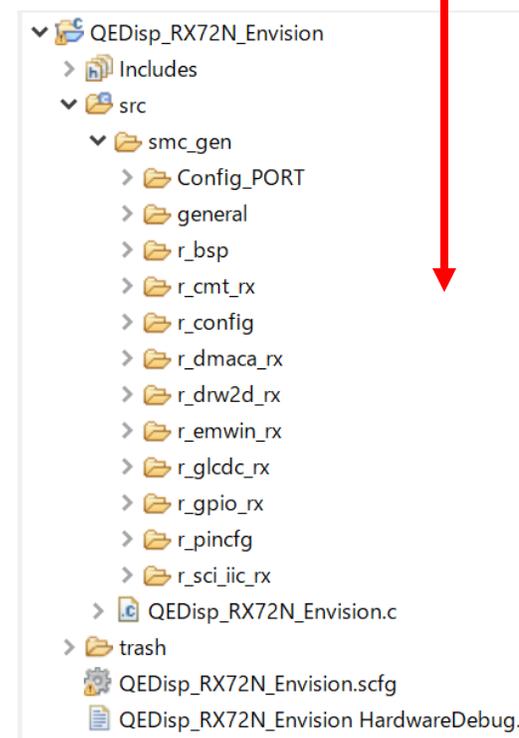
- ① [コンポーネントの追加]アイコンをクリックしてダイアログを開きます。
[機能]項目に「グラフィックス」を選択して「r_emwin_rx」コンポーネントを確認します。
- ② 最新版バージョンではない場合は、[最新版のFITドライバとミドルウェアをダウンロードする]をクリックして、最新版のemWinモジュールをインストールして追加します。
※2023年10月現在の最新版は「v6.32a.1.00」です。

プロジェクト作成：LCDコントローラの導入

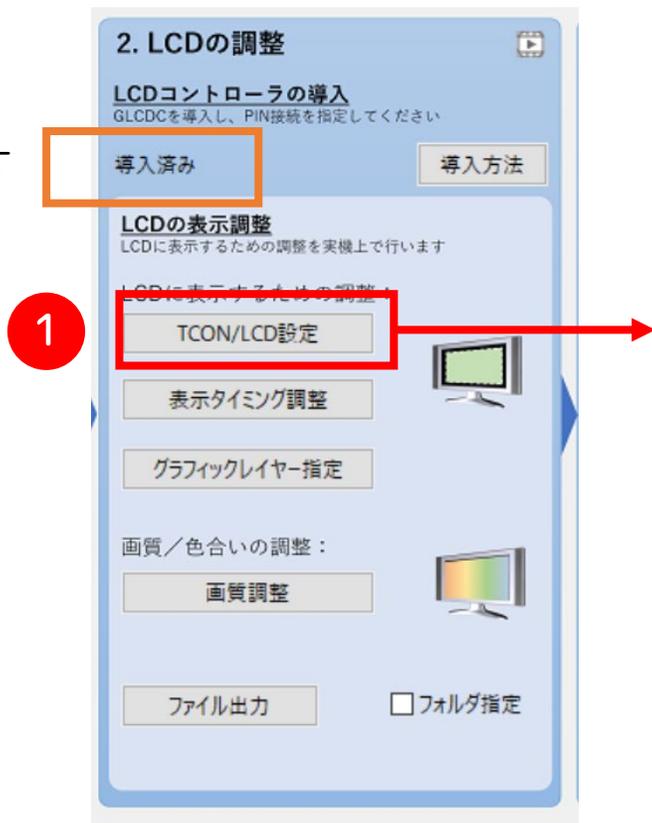
ドライバ追加されましたら、コンポーネントウィンドウに「r_emwin_rx」及び依存するGPIO、CMTドライバモジュールが表示されます。



スマート・コンフィグレータの右上にある「コード生成」ボタンをクリックして、LCDコントローラ関連コードを作成します。



コード作成後に、
「導入済み」となります



① Envision RX72Nボード仕様に合わせてLCD
インターフェースパラメータが設定されています。

プロジェクト作成：LCDの表示調整（表示タイミング調整）

2. LCDの調整

LCDコントローラの導入
GLCDCを導入し、PIN接続を指定してください

導入済み

LCDの表示調整
LCDに表示するための調整を実機上で行います

LCDに表示するための調整：

画質/色合いの調整：

フォルダ指定

ブロック図から選択 TCONとLCD設定 タイミング設定 グラフィックレイヤー設定 画質調整

タイミング調整

PLL回路周波数[MHz]: 240.000000 パネルクロック周波数[MHz]: 10.000000

VPW 1

VBP 7

VDP 272

VFP 8

VTP 288

HPW 14 HBP 75 HDP 480 HFP 15 HTP 584

| | 値 | 規定値 | 差分 |
|---------------|------|------|-----|
| リフレッシュレート[Hz] | 59.5 | 59.5 | 0.0 |
| 水平周波数[kHz] | 17.1 | 17.1 | 0.0 |

Envision RX72NボードのLCD
パネルインターフェースに
合わせて設定します。

パネルクロック周波数 = 10MHz
VPW = 1
VBP = 7
VFP = 8
HPW = 14
HBP = 75
HFP = 15

プロジェクト作成：LCDの表示調整（その他の設定）

2. LCDの調整

LCDコントローラの導入
GLCDCを導入し、PIN接続を指定してください

導入済み 導入方法

LCDの表示調整

LCDに表示するための調整を実機上で行います

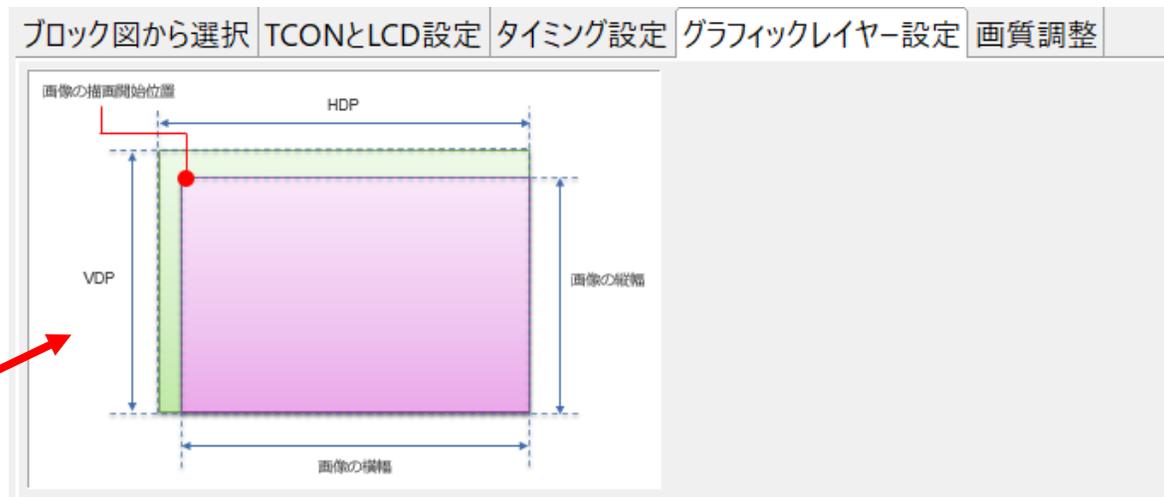
LCDに表示するための調整：

- TCON/LCD設定
- 表示タイミング調整
- グラフィックレイヤー指定
- 画質調整

画質/色合いの調整：

- 画質調整

ファイル出力 フォルダ指定



ブロック図から選択 | TCONとLCD設定 | タイミング調整 | **グラフィックレイヤー設定** | 画質調整

調整処理順設定

調整処理順設定
ブライツネス
コントラスト
ガンマ補正
ディザ処理

簡易設定：
 ブライツネス、コントラスト -> ガンマ補正

カスタム： FRONTGAM

「グラフィックレイヤー設定」、「画質調整」はデフォルトの設定のまま使用します。

プロジェクト作成：LCDの表示調整（ファイル出力）

The screenshot shows a two-step configuration wizard. Step 1, '準備' (Preparation), includes sections for 'プロジェクトの選択' (Project Selection) with 'QEDisp_RX72N_Envision' selected, '評価ボードの選択' (Evaluation Board Selection) with 'EnvisionKitRX72N (V1.12)', 'LCDの選択' (LCD Selection) with 'EastRising', and 'GUI描画ツールの選択' (GUI Drawing Tool Selection) with 'emWin'. Step 2, '2. LCDの調整' (LCD Adjustment), includes 'LCDコントローラの導入' (GLCDC import), 'LCDの表示調整' (LCD display adjustment) with buttons for 'TCON/LCD設定', '表示タイミング調整', and 'グラフィックレイヤー指定', and '画質/色合いの調整' (Quality/Color adjustment) with a '画質調整' button. A red box highlights the 'ファイル出力' (File Output) button at the bottom of the LCD adjustment section, and a red arrow points from it to the file explorer.

The screenshot shows a file explorer window titled 'プロジェクト・エクスプローラー' (Project Explorer) displaying the directory structure of 'QEDisp_RX72N_Envision'. The 'src' folder is expanded, showing subfolders 'smc_gen' and 'trash', and files 'QEDisp_RX72N_Envision.c', 'r_image_config.h', and 'r_lcd_timing.h'. A red box highlights these two header files, and a red arrow points from the 'ファイル出力' button in the previous screenshot to this box. To the right, the 'ソフトウェアコンポーネント設定' (Software Component Settings) window is open, showing a tree view of components like 'Startup', 'ジェネリック', 'Drivers', and '入出力ポート'. Below the file explorer, a 'コンソール' (Console) window shows the output: 'QE for Display' and 'ファイルを生成しました。' (Files generated), followed by two file paths: (1) C:\Workspace\...r_lcd_timing.h and (2) C:\Workspace\...r_image_config.h.

[ファイル出力]ボタンをクリックして、タイミングと画質調整設定ファイルを作成します。

プロジェクト作成：画面表示の作成（emWinの初期設定）

3. 画面表示の作成

GUI描画ツールの導入
GUI描画ツールを導入してください

導入済み

GUI描画ツールの初期設定
GUI描画ツールを使用するための設定を行います

GUI描画ツールの設定： **1**

フォルダ指定

GUIの作成
LCDに表示するGUIを作成します

実装
GUIをディスプレイに表示するコードを、main()関数に組み込みます

emWinの設定

基本情報
使用ボード: RX72N_ENVISION_KIT
画面サイズ(横): 480
画面サイズ(縦): 272

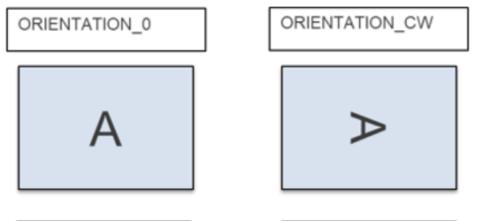
画面の基本設定
LCDの向き: ORIENTATION_0
ピクセル当たりのカラー深度: 16bits

端子設定
 LCDパネルのリセット端子 ポート: B ビット: 3
 LCDパネルのバックライト端子 ポート: 6 ビット: 7
 LCDパネル搭載のタッチICのリセット端子 ポート: 6 ビット: 6

メモリ設定
フレームバッファ1アドレス: 0x00800000
フレームバッファ2アドレス: 0x00840000
GUIで使用する最大メモリサイズ: 81920

タッチ
タッチ機能の使用: 使用する
タッチ接続インタフェース: SCI_IIC
タッチインタフェースのチャネル番号: 6
タッチパネルのスレーブアドレス: 0x38
マルチタッチ機能の使用: 使用しない
タッチパネルのポイントの最大数: 10

DRW2D機能
DRW2Dの使用: 使用する



画面サイズ：480 x 272 px
ビデオフレームバッファLayer1アドレス：0x00800000
ビデオフレームバッファLayer2アドレス：0x00840000
emWin用メモリプールサイズ：81920 Bytes
タッチ機能の使用：使用する
マルチタッチ機能：使用しない
（シングルタッチモード）
DRW2D機能：使用する

- ① [情報設定]ボタンをクリックして設定内容を確認します。
- ② 「OK」ボタンをクリックして設定を保存します。

プロジェクト作成：画面表示の作成（emWin関連ファイル出力）

The screenshot shows the '3. 画面表示の作成' (3. GUI Creation) window in e2Studio. The 'GUI描画ツールの初期設定' (Initial Settings of GUI Drawing Tool) section is active. The 'ファイル出力' (File Output) button is highlighted with a red box and a red arrow pointing to the 'qe_emwin_config.h' file in the project explorer. The console window shows the output of the file generation process.

3. 画面表示の作成

GUI描画ツールの導入
GUI描画ツールを導入してください

導入済み

GUI描画ツールの初期設定
GUI描画ツールを使用するための設定を行います

GUI描画ツールの設定：
 フォルダ指定

GUIの作成
LCDに表示するGUIを作成します

実装
GUIをディスプレイに表示するコードを、main()関数に組み込みます

プロジェクト・エクスプローラー

- QEDisp_RX72N_Envision
 - Includes
 - src
 - smc_gen
 - qe_emwin_config.h
 - QEDisp_RX72N_Envision.c
 - r_image_config.h
 - r_lcd_timing.h
 - trash
 - QEDisp_RX72N_Envision.scfg
 - QEDisp_RX72N_Envision HardwareDebug.laur
 - Developer Assistance

ソフトウェアコンポーネント設定

コンポーネント

フィルタ入力

- Startup
 - ジェネリック
 - r_bsp
- Drivers
 - DMA
 - r_dmaca_rx
 - 入出力ポート
 - Config_PORT

プロパティ

- # DMAC channel n
- # Initialize DMAC c
- # Select DMA tran:
- Select LCD Interface
 - LCD interface
 - GLCDC Interfa

概要 ボード クロック システム **コンポーネント** 端子 割り込み

コンソール

QE for Display

ファイルを生成しました。
(1) C:\Workspace\e2Studio-v2310\QEDisp_RX72N_Envision\src\r_lcd_timing.h
(2) C:\Workspace\e2Studio-v2310\QEDisp_RX72N_Envision\src\r_image_config.h

ファイルを生成しました。
(1) C:\Workspace\e2Studio-v2310\QEDisp_RX72N_Envision\src\qe_emwin_config.h

- ① [ファイル出力]ボタンをクリックして初期設定を反映したヘッダファイル「qe_emwin_config.h」を出力します。

プロジェクト作成： I2Cドライバの設定（タッチパネルI/F用）

ソフトウェアコンポーネント設定

コンポーネント

フィルタ入力

- Startup
 - ジェネリック
 - r_bsp
 - Drivers
 - DMA
 - r_dmaca_rx
 - 入出力ポート
 - Config_PORT
 - r_gpio_rx
 - 通信
 - r_sci_iic_rx**
 - グラフィックス
 - r_drw2d_rx
 - r_glcdc_rx
 - タイマ
 - r_cmt_rx

| プロパティ | 値 |
|---|------------------|
| Configurations | |
| # Set parameter checking enable | Include |
| # MCU supported channels for CH0 | Not supported |
| # MCU supported channels for CH1 | Not supported |
| # MCU supported channels for CH2 | Not supported |
| # MCU supported channels for CH3 | Not supported |
| # MCU supported channels for CH4 | Not supported |
| # MCU supported channels for CH5 | Not supported |
| # MCU supported channels for CH6 | Supported |
| # MCU supported channels for CH7 | Not supported |
| # MCU supported channels for CH8 | Not supported |
| # MCU supported channels for CH9 | Not supported |
| # MCU supported channels for CH10 | Not supported |

Macro definition: SCI_IIC_CFG_CH6_INCLUDED
Selectable whether to use available channels.
0 = Not supported.

emWinの設定

- LCDパネルのリセット端子 ポート: P ピン: P
- LCDパネルのバックライト端子 ポート: 6 ビット: 7
- LCDパネル搭載のタッチICのリセット端子 ポート: 6 ビット: 6

メモリ設定

- フレームバッファ1アドレス: 0x00800000
- フレームバッファ2アドレス: 0x00840000
- GUIで使用する最大メモリサイズ: 81920

タッチ

- タッチ機能の使用: 使用する
- タッチ接続インターフェース: SCI_IIC
- タッチインターフェースのチャンネル番号: 6
- タッチパネルのスレーブアドレス: 0x38
- マルチタッチ機能の使用: 使用しない

Envision RX72NボードのタッチパネルはIICチャンネル「CH6」を使用しています。

emWinでタッチパネルインターフェースに使用するIICチャンネルを指定します。

① スマート・コンフィグレータの右上にある「コード生成」ボタンをクリックして、関連コードを作成します。

The screenshot shows the Smart Configurator interface for a project named *QEDisp_RX72N_Envision.scfg. The main window is titled 'ソフトウェアコンポーネント設定' (Software Component Settings). On the right side of the main window, there are two buttons: 'コードの生成' (Code Generation) and 'レポートの生成' (Report Generation). The 'コードの生成' button is highlighted with a red box and a red circle containing the number '1'. A red arrow points from this button to the 'src' directory in the project's file explorer on the right. The file explorer shows a tree structure under 'QEDisp_RX72N_Envision' with sub-directories like 'Includes' and 'src'. Inside 'src', there is a sub-directory 'smc_gen' which contains several sub-directories and files, including 'Config_PORT', 'general', 'r_bsp', 'r_cmt_rx', 'r_config', 'r_dmaca_rx', 'r_drw2d_rx', 'r_emwin_rx', 'r_glcdc_rx', 'r_gpio_rx', 'r_pinconf', 'r_sci_iic_rx', 'qe_emwin_config.h', 'QEDisp_RX72N_Envision.c', 'r_image_config.h', and 'r_lcd_timing.h'.

| プロパティ | 値 |
|-----------------------------------|---------------|
| ▼ Configurations | |
| # Set parameter checking enable | Include |
| # MCU supported channels for CH0 | Not supported |
| # MCU supported channels for CH1 | Not supported |
| # MCU supported channels for CH2 | Not supported |
| # MCU supported channels for CH3 | Not supported |
| # MCU supported channels for CH4 | Not supported |
| # MCU supported channels for CH5 | Not supported |
| # MCU supported channels for CH6 | Supported |
| # MCU supported channels for CH7 | Not supported |
| # MCU supported channels for CH8 | Not supported |
| # MCU supported channels for CH9 | Not supported |
| # MCU supported channels for CH10 | Not supported |

プロジェクト作成：emWin関連出力ファイルの確認

- ▼ src
 - ▼ smc_gen
 - > Config_PORT
 - > general
 - > r_bsp
 - > r_cmt_rx
 - > r_config
 - > r_dmaca_rx
 - > r_drw2d_rx
 - ▼ r_emwin_rx
 - > doc
 - ▼ lib
 - > Config
 - > GUI
 - > src
 - > tool
 - > r_emwin_rx_if.h
 - readme.txt
 - > r_glcdc_rx
 - > r_gpio_rx
 - > r_pincfg
 - > r_sci_iic_rx
 - > qe_emwin_config.h
 - > QEDisp_RX72N_Envision.c
 - > r_image_config.h
 - > r_lcd_timing.h

| r_emwin_rx フォルダ | 内容詳細 |
|-----------------------|--|
| 「lib」 → 「Config」 フォルダ | GUIConf.c：GUIコンフィグレーション設定ファイル LCDConf.c：LCDドライバ設定ファイル PIDConf.c：タッチパネルインターフェースドライバ |
| 「lib」 → 「GUI」 フォルダ | emWinライブラリファイル その他プロジェクトビルドに必要なヘッダ定義ファイル |
| 「tool」 フォルダ | AppWizardインストールプログラム、画像変換ツール、フォント変換ツール、その他PCツール |

e2Studio-v2310 > QEDisp_RX72N_Envision > src > smc_gen > r_emwin_rx > lib > GUI

GUIの検索

| | | | | | | |
|------------------------|---------------------|---------------------|--------------------|----------------------------|--------------------------|--------------------|
| emWinLib_RXv3_CCRX.lib | SWIPELIST_Private.h | PROGBAR_Private.h | LCD_SIM.h | GUIDRV_Null.h | GUI_Debug.h | DROPDOWN_Private.h |
| WM_Intern.h | SWIPELIST.h | PROGBAR.h | LCD_Protected.h | GUIDRV_Lin.h | GUI_ConfDefaults.h | DROPDOWN.h |
| WM_GUI.h | SPINBOX_Private.h | MULTIPAGE_Private.h | LCD_Private.h | GUIDRV_FlexColor_Private.h | GUI_BMP_Private.h | DIALOG_Type.h |
| WM.h | SPINBOX.h | MULTIPAGE.h | LCD_ConfDefaults.h | GUIDRV_FlexColor.h | GUI_ARRAY_Private.h | DIALOG.h |
| WINDOW_Private.h | SLIDER_Private.h | MULTIEDIT_Private.h | LCD.h | GUI_VNC.h | GUI_ARRAY.h | CHOOSEFILE.h |
| WIDGET_ID.h | SLIDER.h | MULTIEDIT.h | KNOB_Private.h | GUI_Version.h | GUI_BiDi2_Types_800.h | CHOOSECOLOR.h |
| WIDGET.h | SEGGGER.h | MOVIE_Private.h | KNOB.h | GUI_Type.h | GUI_BiDi2_Brackets_800.h | CHECKBOX_Private.h |
| WHEEL_Private.h | SCROLLER_Private.h | MOVIE.h | KEYBOARD_Private.h | GUI_SPY_Private.h | GUI.h | CHECKBOX.h |
| WHEEL.h | SCROLLER.h | MESSAGEBOX.h | KEYBOARD.h | GUI_SPRITE_Private.h | GRAPH_Private.h | CALENDAR.h |
| TREEVIEW_Private.h | SCROLLBAR_Private.h | MENU_Private.h | IP_FS.h | GUI_SetOrientationCX.h | GRAPH.h | BUTTON_Private.h |
| TREEVIEW.h | SCROLLBAR.h | MENU.h | IMAGE_Private.h | GUI_SetOrientation.h | Global.h | BUTTON.h |
| TICKER_Private.h | ROTARY_Private.h | LISTWHEEL_Private.h | IMAGE.h | GUI_Private.h | GAUGE_Private.h | AppWizard.h |
| TICKER.h | ROTARY.h | LISTWHEEL.h | ICONVIEW_Private.h | GUI_JPEG_Private.h | GAUGE.h | |
| TEXT_Private.h | RADIO_Private.h | LISTVIEW_Private.h | ICONVIEW.h | GUI_HOOK.h | FRAMEWIN_Private.h | |
| TEXT.h | RADIO.h | LISTVIEW.h | HEADER_Private.h | GUI_GIF_Private.h | FRAMEWIN.h | |
| SWITCH_Private.h | QRCODE_Private.h | LISTBOX_Private.h | HEADER.h | GUI_GCache_Private.h | EDIT_Private.h | |
| SWITCH.h | QRCODE.h | LISTBOX.h | GUIDRV_Templatel.h | GUI_FontIntern.h | EDIT.h | |

プロジェクト作成：ビルド確認

The screenshot shows the e2Studio IDE interface. The top menu bar includes 'ファイル(F)', '編集(E)', 'Source', 'ナビ', '検索(A)', 'プロジェクト(P)', 'Renesas Views', '実行(R)', 'Renesas AI', 'ウィンドウ(W)', and 'ヘルプ(H)'. The main workspace is divided into several panes:

- Left Pane:** Project Explorer showing the project structure for 'QEDisp_RX72N_Envision'. A red box highlights the project name, and a red circle with the number '2' is next to it.
- Center Pane:** 'デバイス選択' (Device Selection) dialog. It shows 'ボード: EnvisionKitRX72N (V1.13)' and 'デバイス: R5F572NNHxFB'. A red circle with the number '1' is next to the 'デバイス' field.
- Right Pane:** 'MCU/MPUパッケージ' (MCU/MPU Package) and 'Developer Assist Browser' showing a Renesas RX72N chip.
- Bottom Left Pane:** 'コンソール' (Console) showing build output. A red box highlights the final line: '17:12:39 Build Finished. 0 errors, 0 warnings. (took 56s.723ms)'. Above this, it shows 'Linking start' and 'Building target:'.
- Bottom Right Pane:** 'コンフィギュレーションチェック' (Configuration Check) showing a checklist of steps: 1. 準備 (Preparation), 2. LCDの調整 (LCD Adjustment), 3. 画面表示の作成 (Screen Display Creation). Each step has a '導入方法' (Import Method) button.

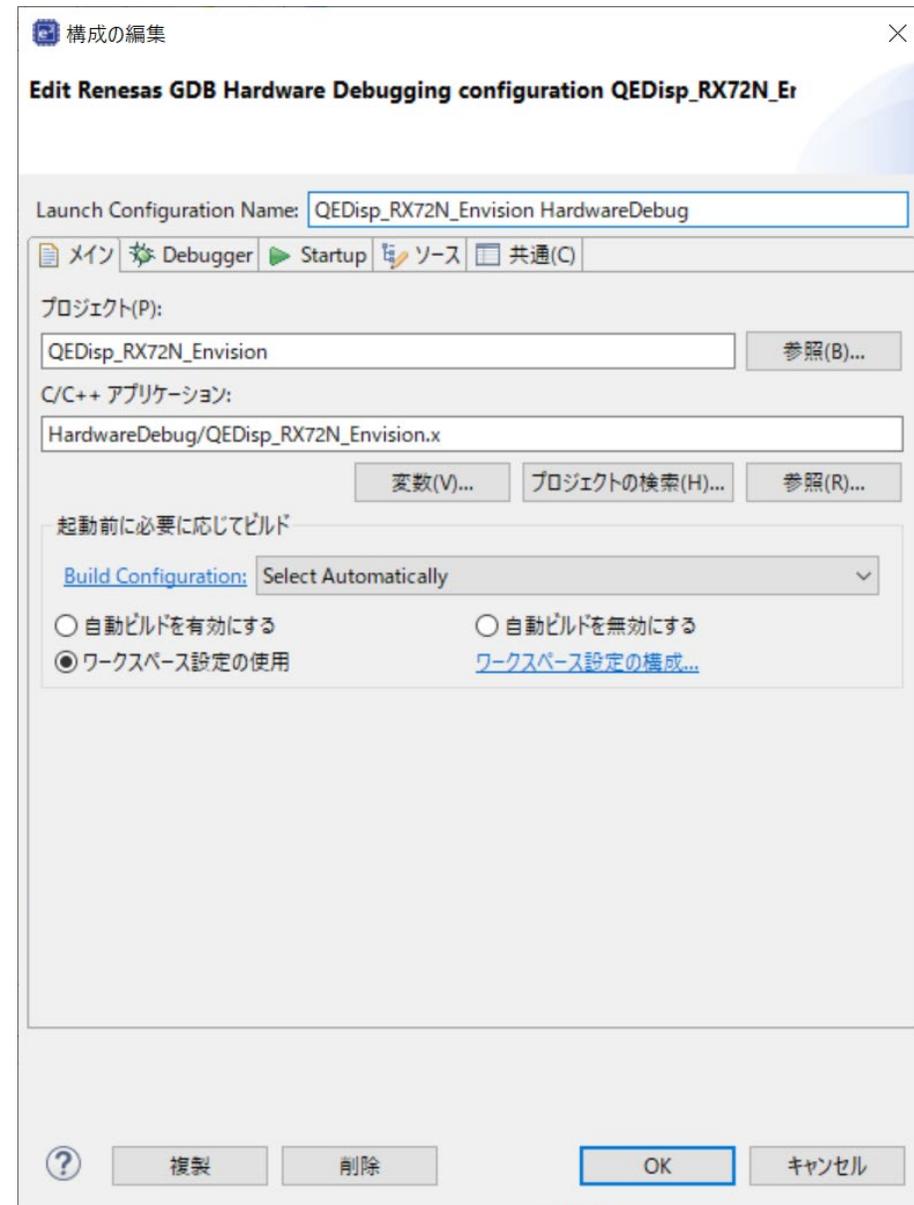
プロジェクトをビルドしてファイル・モジュールが不足していないことを確認します。不足しているコンポーネントを追加してください。

プロジェクト作成：デバッグ設定

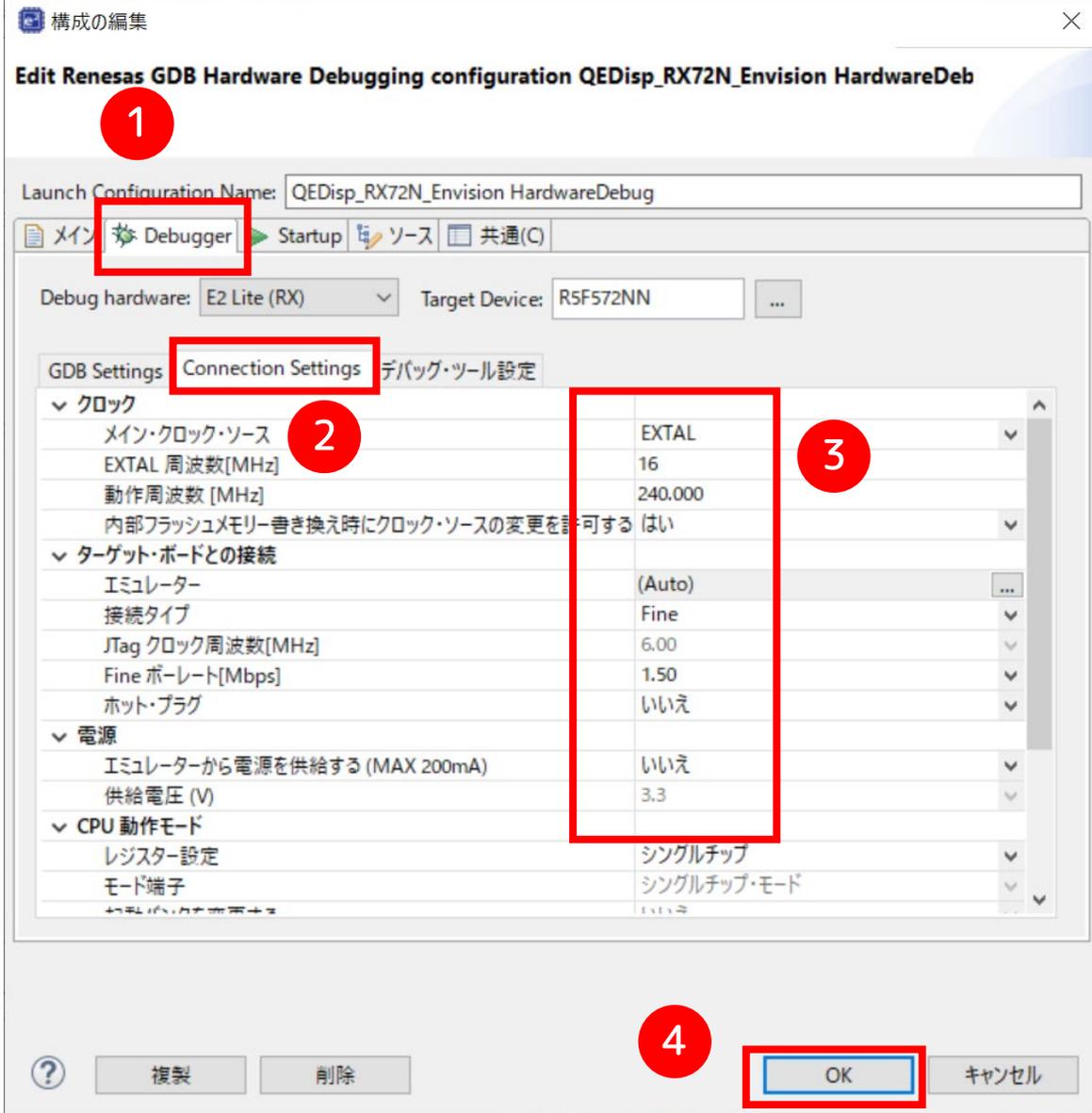


① デバッグするプロジェクト [QEDisp_RX72N_Envision] タイトルをクリックして選択

② メニューの [] アイコンをクリックしてデバッグ構成編集ダイアログを開きます



プロジェクト作成：デバッグ設定



- ① [Debugger] → ② [Connection Settings] タブを選択します。
- ③ 設定項目を確認します。

| | |
|------------------|---------|
| メイン・クロック・ソース | EXTAL |
| EXTAL 周波数 [MHz] | 16.0000 |
| 接続タイプ | Fine |
| エミュレーターから電源を供給する | いいえ |

- ④ [OK] ボタンをクリックして設定を保存します。

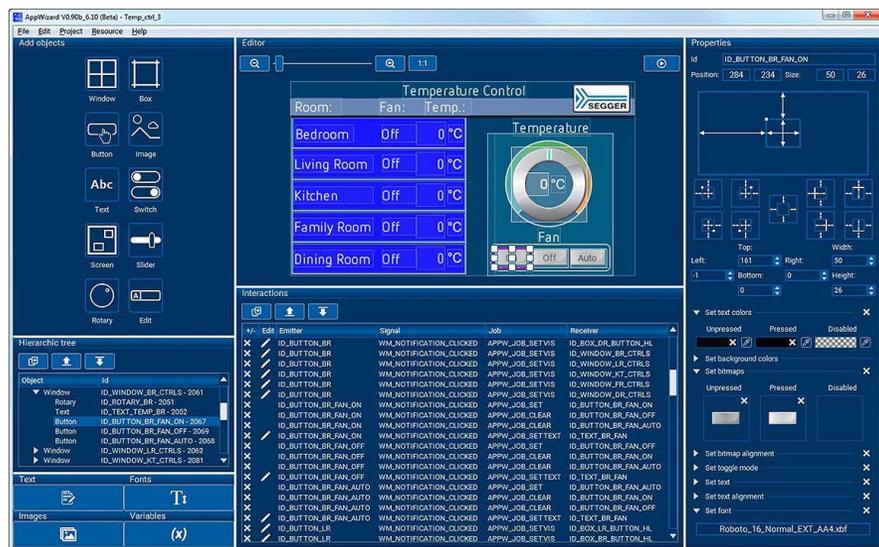
TASK-3

画面表示の作成 (事前設定)

emWinアプリケーションのスタートアップコードを設定します。

オプション1

GUI描画ツールAppWizardから画面をデザインしてemWinアプリケーションコードを作成します。



※ AppWizardツールのインストールが必要です。

オプション2

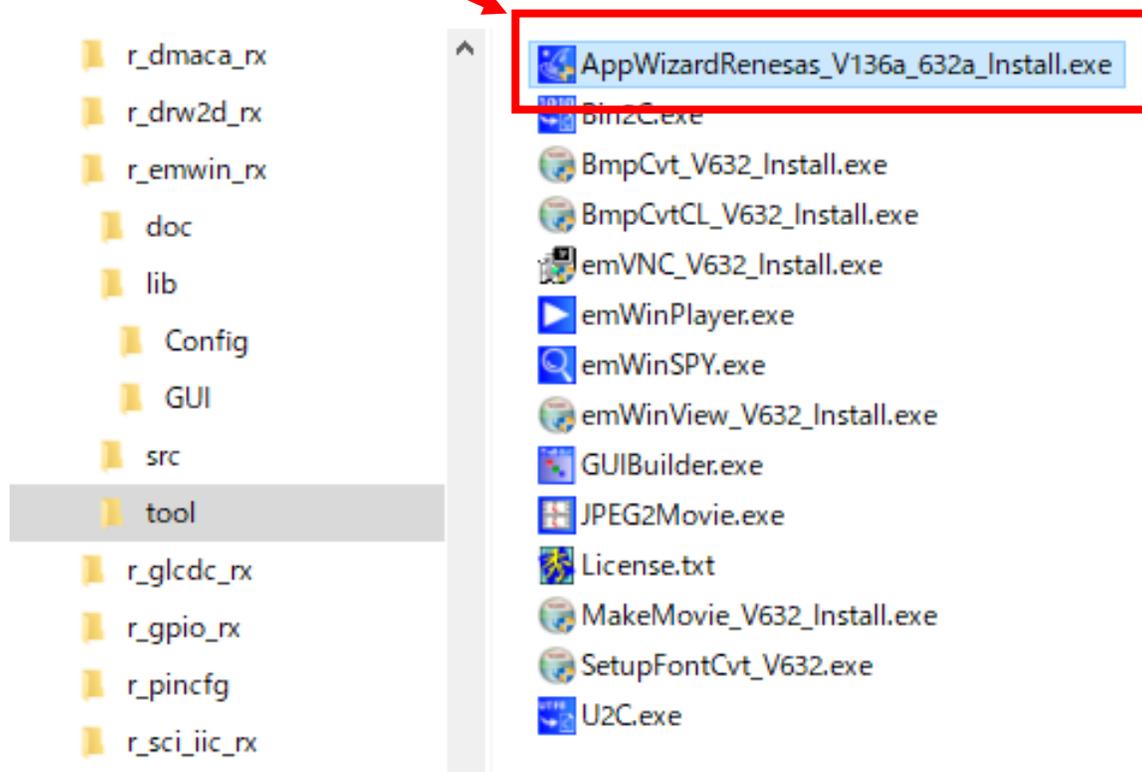
AppWizardなしでウィジェット及びemWin APIを使用してGUIアプリケーションを作成して画面を実装します。



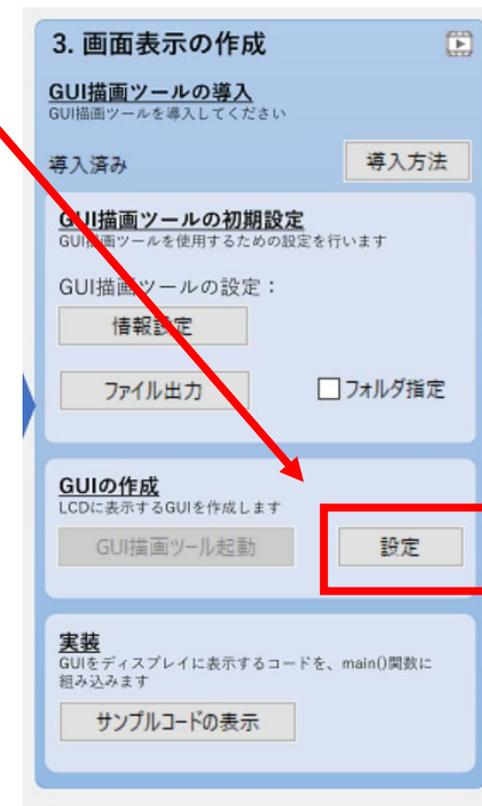
※ 必要なツール（フォント変換、ビットマップ変換ツールなど）はemWinパッケージに含まれています。

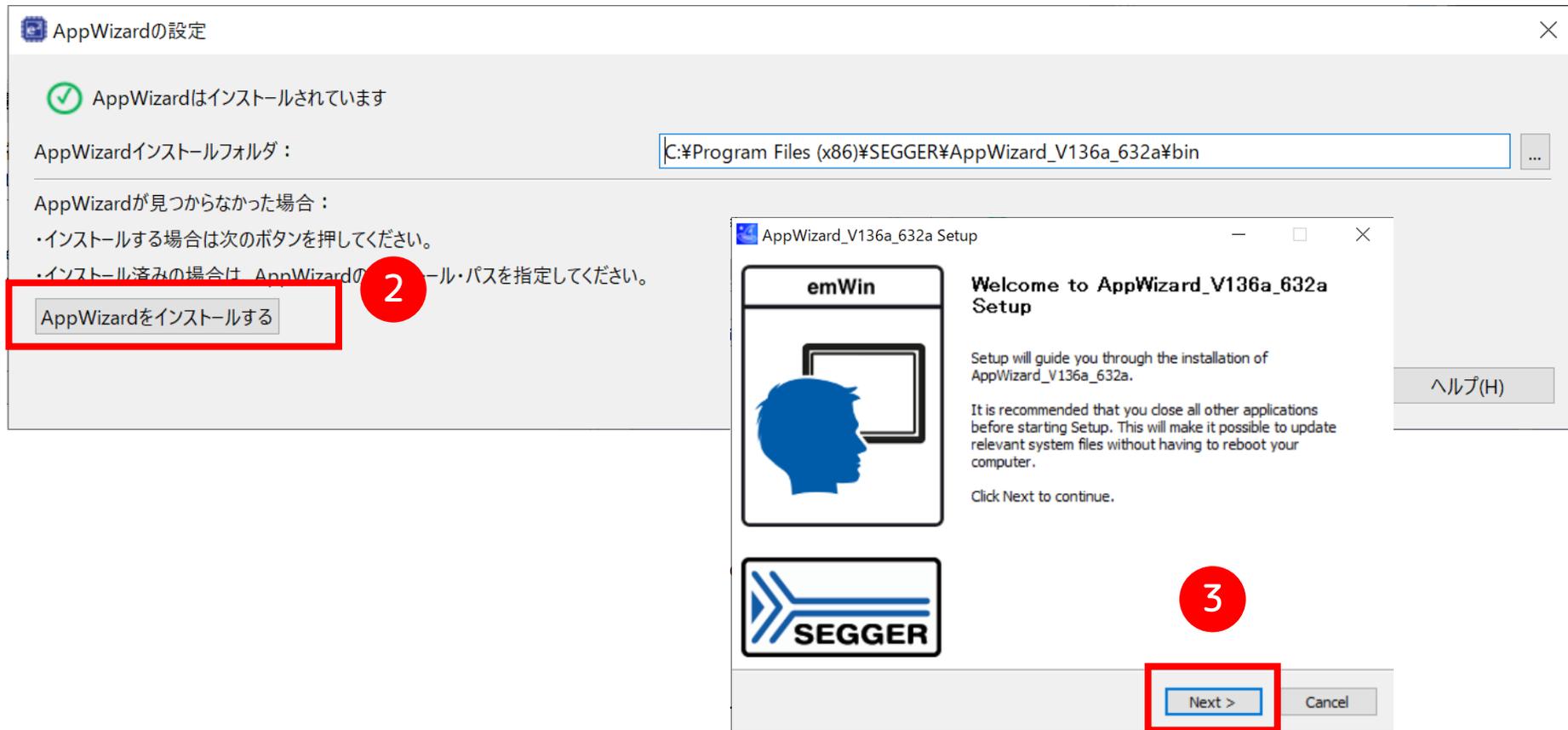
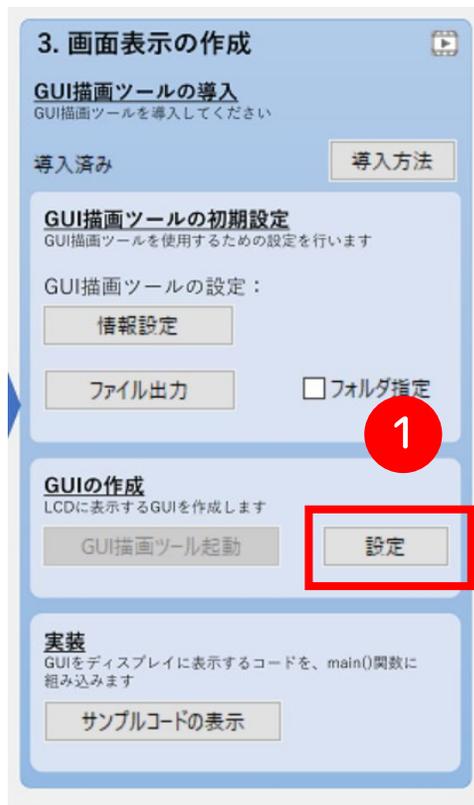
AppWizardツールのインストーラはスマート・コンフィグレータが作成する「r_emwin_rx → tool」フォルダにあります。
AppWizardインストーラ起動方法は2つあります。

方法1：「r_emwin_rx → tool」フォルダ下のEXEを実行する



方法2：e2Studio → QE for DisplayからAppWizardインストーラを起動する

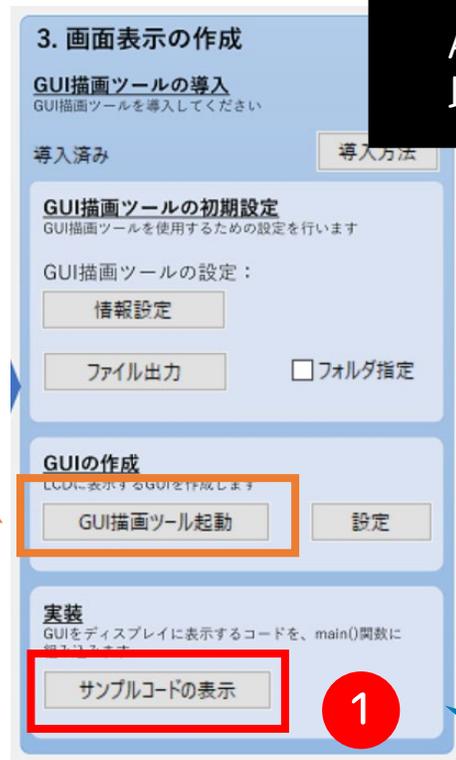




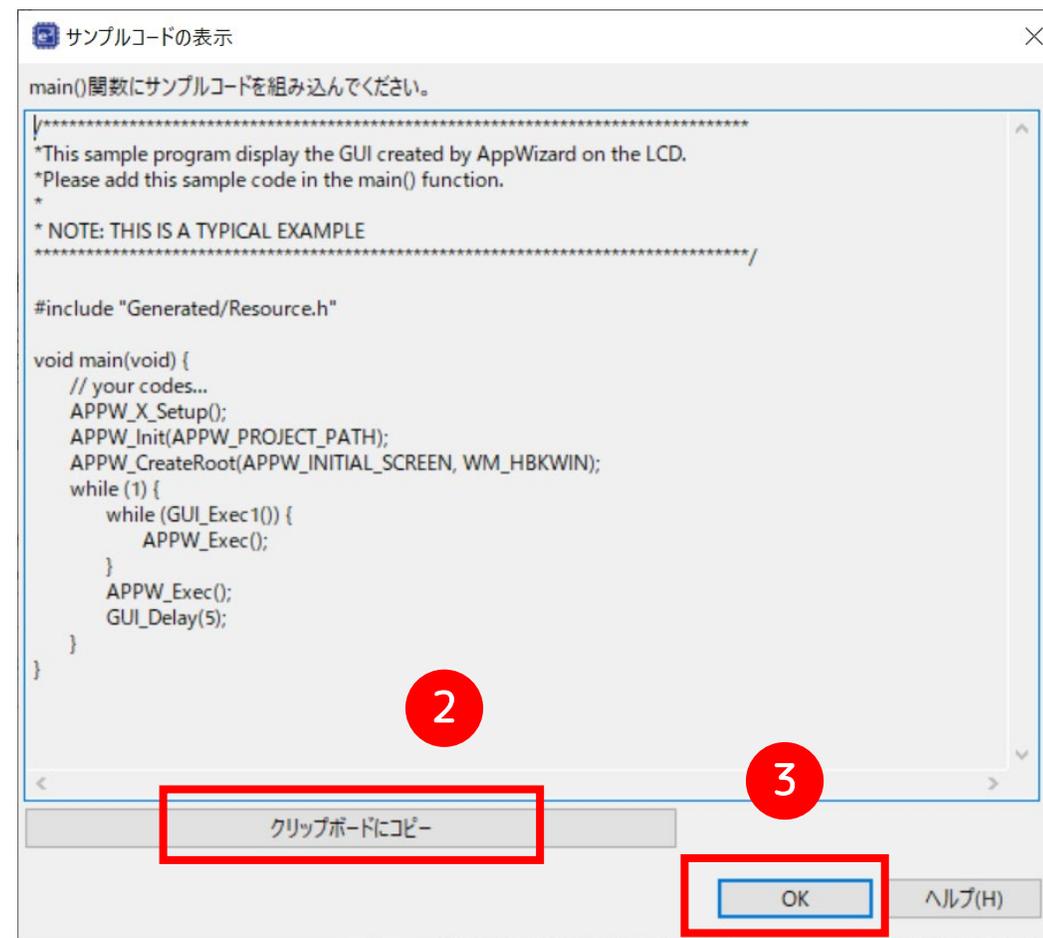
メニューから[Renesas Views]→[Renesas QE] →[LCD メインRX (QE)]コマンドを実行します。

- ① [GUI の作成]の[設定]ボタンをクリックして[AppWizard の設定]ダイアログを開きます。
- ② [AppWizard をインストールする]ボタンをクリックしてインストーラを起動します。
- ③ インストーラ実行を進みます。

AppWizard のインストール後に、
[GUI 描画ツール起動]ボタンは
有効になります。



AppWizardツールで画面アプリケーションコードを作成しましたら、
以下の手順でemWinスタートアップ・起動処理を設定します。



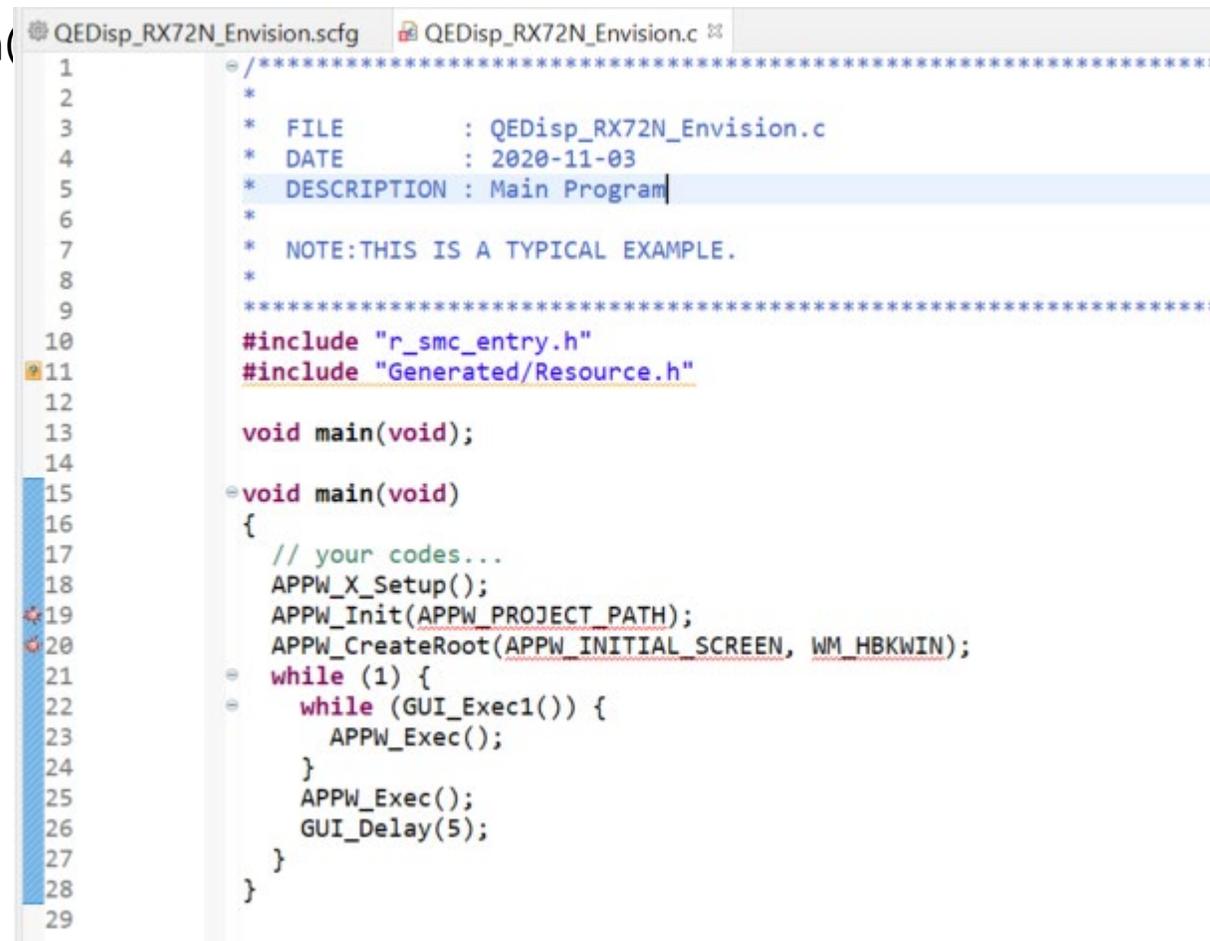
- ① QE for Display [RX]の[3. 画面表示の作成]下部の[サンプルコードの表示]ボタンをクリックします。
- ② [サンプルコードの表示]ダイアログで[クリップボードにコピー]ボタンをクリックして、③[OK]ボタンをクリックしてダイアログを閉じます。

AppWizardツールで画面アプリケーションコードを作成しましたら、以下の手順でemWinスタートアップ・起動処理を設定します。

コピーしたコードをアプリケーションプログラムのmain関数内に埋め込みます。

```
#include "Generated/Resource.h"
```

```
void main(void) {  
    // your codes...  
    APPW_X_Setup();  
    APPW_Init(APPW_PROJECT_PATH);  
    APPW_CreateRoot(APPW_INITIAL_SCREEN, WM_HBKWIN);  
    while (1) {  
        while (GUI_Exec1()) {  
            APPW_Exec();  
        }  
        APPW_Exec();  
        GUI_Delay(5);  
    }  
}
```



```
QEDisp_RX72N_Envision.scfg QEDisp_RX72N_Envision.c  
1  /*  
2  *  
3  * FILE      : QEDisp_RX72N_Envision.c  
4  * DATE      : 2020-11-03  
5  * DESCRIPTION : Main Program  
6  *  
7  * NOTE: THIS IS A TYPICAL EXAMPLE.  
8  *  
9  *  
10 *****  
11 #include "r_smc_entry.h"  
12 #include "Generated/Resource.h"  
13  
14 void main(void);  
15  
16 void main(void)  
17 {  
18     // your codes...  
19     APPW_X_Setup();  
20     APPW_Init(APPW_PROJECT_PATH);  
21     APPW_CreateRoot(APPW_INITIAL_SCREEN, WM_HBKWIN);  
22     while (1) {  
23         while (GUI_Exec1()) {  
24             APPW_Exec();  
25         }  
26         APPW_Exec();  
27         GUI_Delay(5);  
28     }  
29 }
```

AppWizardなしでウィジェットおよびemWin APIでGUIアプリケーションは作成可能です。

必要なツール（フォント変換、画像・動画変換ツールなど）はe2Studioから作成する「r_emwin_rx → tool」フォルダ下のEXEプログラムをご使用ください。

emWin APIで実装するGUIアプリケーションの場合は、以下のコードをアプリケーションプログラムのmain()関数内に埋め込みます。

```
#include "GUI.h" ← emWinヘッダ定義
#include "Dialog.h"

void main(void) {
    // your codes...
    GUI_Init(); ← GUIライブラリの初期化
    ...
    while (1) {
        ...
        GUI_Exec(); ← emWinウィンドウマネージャ
        GUI_Delay(5);
        処理の実行
    }
}
```

```
QEDisp_RX72N_Envision.scfg  QEDisp_RX72N_Envision.c
/
*
* FILE      : QEDisp_RX72N_Envision.c
* DATE     : 2020-11-03
* DESCRIPTION : Main Program
*
* NOTE: THIS IS A TYPICAL EXAMPLE.
*
*****
#include "r_smc_entry.h"
#include "GUI.h"
#include "Dialog.h"

void main(void);

void main(void)
{
    // your codes...
    GUI_Init();
    GUI_SetBkColor(GUI_RED);
    GUI_Clear();
    GUI_SetTextMode(GUI_TM_TRANS);
    GUI_SetFont(GUI_FONT_24B_ASCII);
    GUI_DispString("Hello world!");
    for(;;)
    {
        GUI_Exec();
        GUI_Delay(1);
    }
}
```

100種を超えるemWin APIサンプルアプリケーション・ソースコードを提供しています。

emWin Tutorials

Download executables of the emWin Tutorials package [zip archive]

Download 7z archive

Download source code of the emWin Tutorials package [zip archive]

Download 7z archive

- Windows PC環境で実行できるデモサンプル（emWin API用）：

https://www.segger.com/downloads/emwin/emWin_Tutorials_EXE

- チュートリアル・デモサンプルソースコード（emWin API用）：

https://www.segger.com/downloads/emwin/emWin_Tutorials_SRC

- オンラインデモサンプル（emWin API用）：

https://wiki.segger.com/emWin_Examples



- RX72N Envision Kitの詳細

<https://www.renesas.com/jp/ja/products/software-tools/boards-and-kits/eval-kits/rx72n-envision-kit.html>

- RX65N / RX72N Envision Kit用emWinスタートアップガイド (PDF)

https://www.embitek.co.jp/technote/emwin/StartupGuide_emWin_EnvisionKit.pdf

- RXファミリ QE for Display [RX] アプリケーションノート

<https://www.renesas.com/jp/ja/document/apn/rx-family-qe-display-rx-application-note>

- emWinユーザガイド (オンライン)

https://www.segger.com/doc/UM03001_emWin.html

- AppWizardユーザガイド (オンライン)

https://www.segger.com/doc/UM03003_AppWizard.html

ソースパッケージアップグレードのご案内

OEM版emWinライブラリパッケージをご利用のユーザー様がソース付きのemWin PROパッケージへ**割引**価格でアップグレード可能です。

※ ライセンス製品価格・割引率は、サポート窓口対応や諸条件、為替相場により提示価格が変動する場合があります。

ソースアップグレードのメリット：

- ソース付きプロジェクトでGUIアプリケーションのデバッグ操作が簡単
- ソースを参考に新しいウィジェット作成可能
- emWinライブラリの最適化はさらにチューニング可能
- お困りの際に技術サポートへの問い合わせ可能

お気軽に以下窓口へお問い合わせください。

株式会社エンビテック

代理店販売（デバッガ、RTOS、GUI、ミドルウェア）

TEL: 03-6240-2655 / FAX: 03-6240-2656

E-mail : sales@embitek.co.jp

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

