

## RX ファミリ

### QE for Display [RX] サンプルプログラム

---

#### 要旨

本アプリケーションノートでは、ルネサス製 RX マイコンに対応する統合開発環境 e<sup>2</sup> studio 用のプラグインである“QE for Display [RX]”と連携するサンプルアプリケーションについて説明します。“QE for Display [RX]”は、表示機器を搭載した組み込みシステム開発において、表示制御をグラフィカルな I/F でサポートするツールです。“QE for Display [RX]”を使用してシステム開発を行うためには、RX ファミリに搭載されているグラフィック LCD コントローラ（以下、GLCDC）を初期化するためのプログラムが必要です。本アプリケーションノートでは、GLCDC を初期化するためのベースとなるサンプルプログラムを提供します。

#### 対象デバイス

- ・ RX65N グループ、RX651 グループ      ROM 容量 : 1.5MB ~ 2MB
- ・ RX72N グループ
- ・ RX72M グループ
- ・ RX66N グループ

本アプリケーションノートは、

- ・ Renesas Starter Kit+ for RX72N、
- ・ Renesas Envision KIT RPBRX72N、
- ・ Renesas Starter Kit+ for RX65N-2MB、
- ・ Renesas Envision KIT RPBRX65N

向けのサンプルプログラムを用意しています。

その他のデバイス、ボードに適用する場合は、「10 ユーザ環境に適用するには」を参照してください。

## 目次

1.	概要	4
1.1	QE for Display [RX]を使った開発のフロー	5
2.	動作環境	6
3.	関連ドキュメント	10
4.	サンプルプロジェクトの構成	11
5.	サンプルプロジェクトの実行手順	12
5.1	QE for Display [RX]のインストール	13
5.2	プロジェクトのインポート	14
5.3	プロジェクトのビルド	16
5.4	デバッガ接続とプログラムの実行	17
5.5	QE for Display [RX]によるリアルタイム調整	19
6.	ハードウェア説明	20
6.1	ハードウェア構成	20
6.2	端子機能	20
7.	ソフトウェア説明	23
7.1	動作概要	23
7.2	GLCDC の動作設定詳細	24
7.2.1	パラメータ設定(QE)	25
7.2.2	パラメータ設定(ユーザ)	25
7.2.3	GLCDC の端子設定	25
7.2.4	ボードの端子設定	25
7.3	GLCDC FIT モジュールのパラメータと QE for Display [RX]が出力するヘッダとの対応	26
7.4	周辺機能	34
7.5	メモリ	34
7.6	メモリマップ	35
7.7	使用割り込み一覧	35
7.8	本サンプルの使用に際して	36
7.8.1	変数一覧	36
7.8.2	関数一覧	37
8.	QE for Display [RX]の使用方法	38
8.1	QE for Display[RX]の起動	38
8.2	LCD パネルデータの設定	39
8.3	制御信号の出力設定	39
8.4	LCD パネル制御信号タイミングの調整	41
8.5	制御信号出力/タイミング調整結果の反映	43
8.6	イメージダウンロード機能	44
8.7	各種画質の調整	46
8.8	画質調整用ヘッダファイルの生成	48
9.	LCD パネルデータの設定詳細	49
9.1	登録名称の記入	49
9.2	表示方式の選択	50
9.3	制御タイミングの入力	51
9.4	作成したディスプレイデータの編集	52
10.	ユーザ環境に適用するには	53
10.1	仕様確認	54
10.1.1	パネルクロック	54

---

10.1.2	端子割り付け(ビットエンディアン)とピクセル配列 .....	54
10.1.3	その他の制御端子 .....	55
10.2	プロジェクトの作成 .....	56
10.2.1	Smart Configurator の選択 .....	56
10.2.2	セクションの設定 .....	57
10.3	Smart Configurator の設定 .....	58
10.3.1	クロックの設定 .....	58
10.3.2	GLCDC FIT モジュールの追加 .....	58
10.3.3	GLCDC で使用する端子の設定 .....	58
10.4	QE for Display [RX]による調整(初期設定) .....	58
10.5	プログラムの作成 .....	59
10.5.1	サンプルプログラムのコピー .....	59
10.5.2	プログラムの変更 .....	59
10.6	実行から調整終了まで .....	60

## 1. 概要

GLCDC は、図 1-1 に示すように複数のブロックで構成されているため、表示確認を行うだけでも GLCDC の仕様を理解していくつもの設定を行う必要があります。そこで、本サンプルプログラムと QE for Display [RX] を使用する事で GLCDC の仕様を理解する必要なく短期間で表示機器の接続確認ができる環境を準備しました。QE for Display [RX] は、表示制御をグラフィカルな I/F でサポートするツールです。使用する表示機器の情報を入力することで、表示制御に必要な情報を含んだヘッダファイルを出力します。そのヘッダファイルをもとにサンプルプログラムが GLCDC の設定を行います。

また、ツールにはリアルタイムでタイミングを調整する機能があり、使用する表示機器を接続したまま微調整を行ってから、ヘッダファイルを出力する事も可能です。

さらに、デバイスの設定をグラフィカルにサポートする Smart Configurator や、RX ファミリのドライバ/ミドルウェアを提供する Firmware Integration Technology (以下、FIT) を使用し、QE for Display [RX] と連携することで、より簡単に表示制御を行うことが可能になります。本サンプルでは、QE for Display [RX] を主体に、Smart Configurator および FIT が提供する“グラフィック LCD コントローラモジュール Firmware Integration Technology” (以下、GLCDC FIT モジュール) を使用しています。

以降に本サンプルについて記載します。

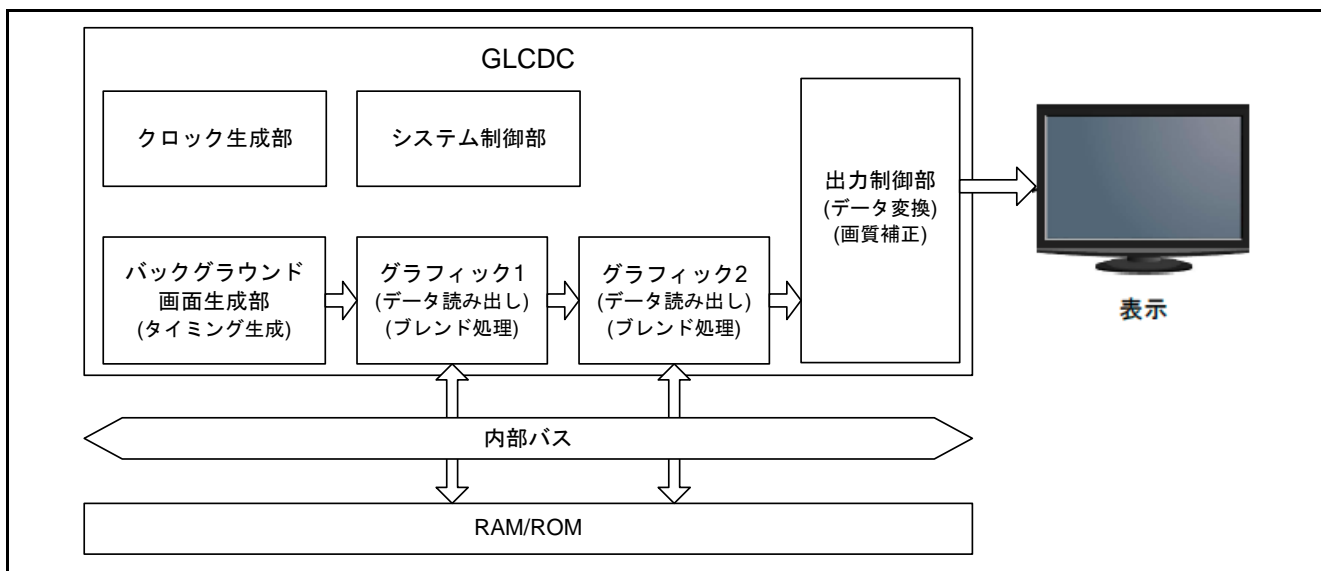


図 1-1 GLCDC のブロック構成

## 1.1 QE for Display [RX]を使った開発のフロー

QE for Display [RX] を使用したシステム開発のフローを図 1-2 に示します。

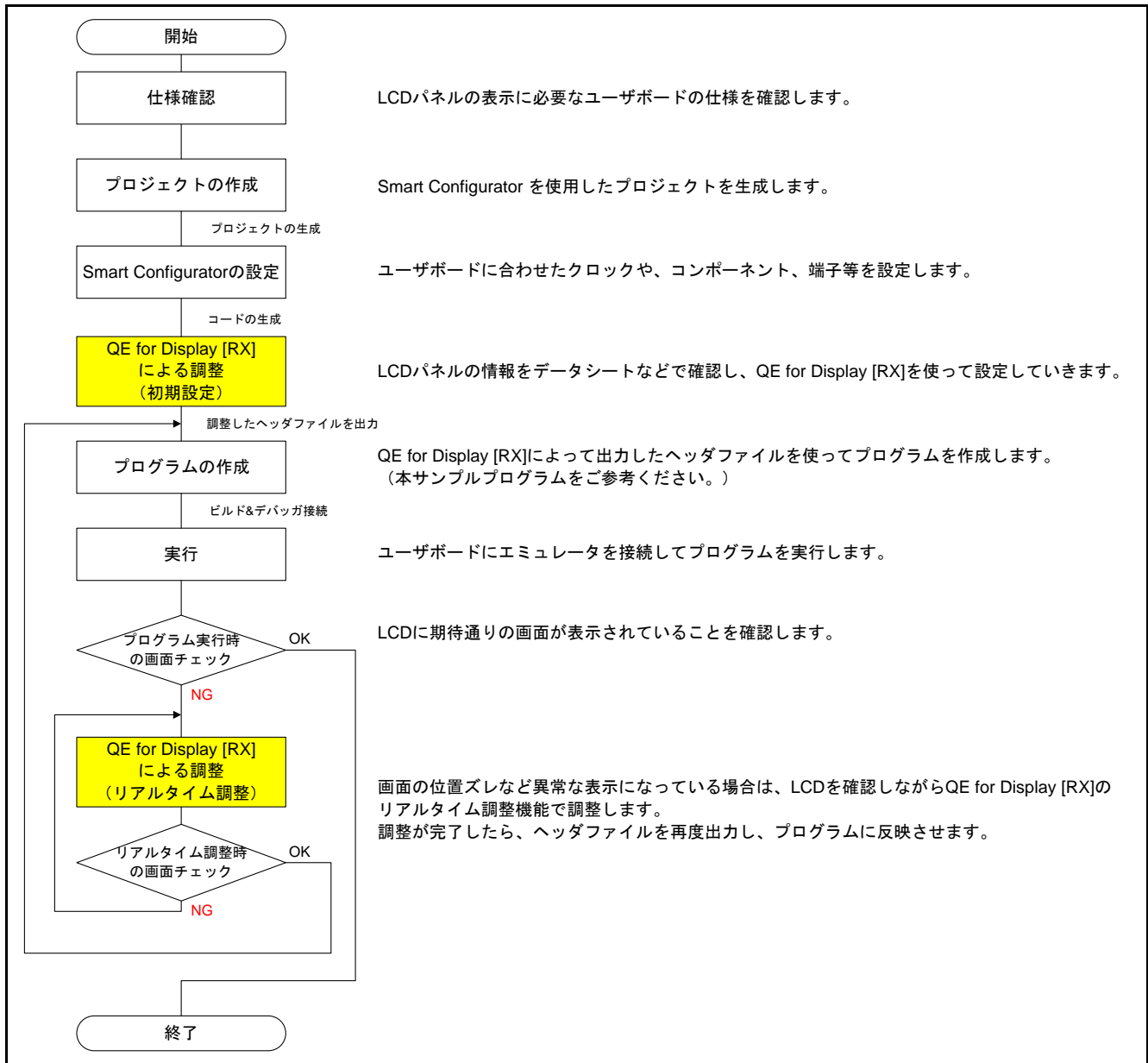


図 1-2 QE for Display [RX]を使用したシステム開発

## 2. 動作環境

本サンプルは、Renesas Starter Kit+ for RX72N（以下、RSK RX72N）、Renesas Envision KIT RPBRX72N（以下、Envision RX72N）、Renesas Starter Kit+ for RX65N-2MB（以下、RSK RX65N）と Renesas Envision KIT RPBRX65N（以下、Envision RX65N）で動作を確認しています。

各ボードの動作確認条件を以下示します。

表 2-1 動作確認条件（RSK RX72N）

項目	内容
使用マイコン	R5F572NNDDBD (RX72N グループ)
動作周波数	<ul style="list-style-type: none"> <li>● メインクロック: 24MHz</li> <li>● PLL: 240MHz (メインクロック 1 分周 10 通倍)</li> <li>● システムクロック (ICLK): 240MHz (PLL 1 分周)</li> <li>● 周辺モジュールクロック A(PCLKA): 120MHz (PLL 2 分周)</li> <li>● 周辺モジュールクロック B(PCLKB): 60MHz (PLL 4 分周)</li> <li>● LCD パネルクロック (LCD_CLK): 10MHz (PLL 24 分周)</li> </ul>
動作電圧	3.3V
統合開発環境	ルネサスエレクトロニクス製 e <sup>2</sup> studio Version 7.7.0
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler Package for RX Family V.3.01.00 コンパイルオプション -lang = c99
iodefine.h のバージョン	Version 1.0
エンディアン	リトルエンディアン、ビッグエンディアン
動作モード	シングルチップモード
プロセッサモード	スーパバイザモード
サンプルプログラムのバージョン	Version 1.10
エミュレータ	E2 Lite
使用ボード	Renesas Starter Kit+ for RX72N（型名：RTK5572NNxxxxxxx）

表 2-2 動作確認条件 (Envision RX72N)

項目	内容
使用マイコン	R5F572NNHDFB (RX72N グループ)
動作周波数	<ul style="list-style-type: none"> <li>● メインクロック: 16MHz</li> <li>● PLL: 240MHz (メインクロック 1 分周 15 通倍)</li> <li>● システムクロック (ICLK): 240MHz (PLL 1 分周)</li> <li>● 周辺モジュールクロック A (PCLKA): 120MHz (PLL 2 分周)</li> <li>● 周辺モジュールクロック B (PCLKB): 60MHz (PLL 4 分周)</li> <li>● LCD パネルクロック (LCD_CLK): 10MHz (PLL 24 分周)</li> </ul>
動作電圧	3.3V
統合開発環境	ルネサスエレクトロニクス製 e <sup>2</sup> studio Version 7.7.0
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler Package for RX Family V.3.01.00
	コンパイルオプション -lang = c99
iodefine.h のバージョン	Version 1.0
エンディアン	リトルエンディアン、ビッグエンディアン
動作モード	シングルチップモード
プロセッサモード	スーパバイザモード
サンプルプログラムのバージョン	Version 1.10
エミュレータ	E2OB (E2 オンボードエミュレータ)
使用ボード	Renesas Envision KIT RPBRX72N (製品型名: RTK5RX72N0CxxxxxBJ)
ボード設定 (ジャンパ / スイッチ)	<SW1> Pin 1: don't care Pin 2: OFF (デバッグを使用する) <その他> デフォルト

表 2-3 動作確認条件 (RSK RX65N)

項目	内容
使用マイコン	R5F565NEDDFC (RX65N グループ)
動作周波数	<ul style="list-style-type: none"> <li>● メインクロック: 24MHz</li> <li>● PLL: 240MHz (メインクロック 1 分周 10 通倍)</li> <li>● システムクロック (ICLK): 120MHz (PLL 2 分周)</li> <li>● 周辺モジュールクロック A(PCLKA): 120MHz (PLL 2 分周)</li> <li>● 周辺モジュールクロック B(PCLKB): 60MHz (PLL 4 分周)</li> <li>● LCD パネルクロック (LCD_CLK): 10MHz (PLL 24 分周)</li> </ul>
動作電圧	3.3V
統合開発環境	ルネサスエレクトロニクス製 e <sup>2</sup> studio Version 7.7.0
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler Package for RX Family V.3.01.00 コンパイルオプション -lang = c99
iodefine.h のバージョン	Version 2.3
エンディアン	リトルエンディアン、ビッグエンディアン
動作モード	シングルチップモード
プロセッサモード	スーパバイザモード
サンプルプログラムのバージョン	Version 1.10
エミュレータ	E2 Lite
使用ボード	Renesas Starter Kit+ for RX65N-2MB (製品型名: RTK50565Nxxxxxxxx)
ボード設定 (ジャンパ / スイッチ)	<SW4> Pin 3:OFF Pin 4:ON (LCD を使用する) <その他> デフォルト



表 2-4 動作確認条件 (Envision RX65N)

項目	内容
使用マイコン	R5F565NEDDFB (RX65N グループ)
動作周波数	<ul style="list-style-type: none"> <li>● メインクロック: 12MHz</li> <li>● PLL: 240MHz (メインクロック 1 分周 20 通倍)</li> <li>● システムクロック (ICLK): 120MHz (PLL 2 分周)</li> <li>● 周辺モジュールクロック A (PCLKA): 120MHz (PLL 2 分周)</li> <li>● 周辺モジュールクロック B (PCLKB): 60MHz (PLL 4 分周)</li> <li>● LCD パネルクロック (LCD_CLK): 10MHz (PLL 24 分周)</li> </ul>
動作電圧	3.3V
統合開発環境	ルネサスエレクトロニクス製 e <sup>2</sup> studio Version 7.7.0
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler Package for RX Family V.3.01.00
	コンパイルオプション -lang = c99
iodefine.h のバージョン	Version 2.3
エンディアン	リトルエンディアン、ビッグエンディアン
動作モード	シングルチップモード
プロセッサモード	スーパバイザモード
サンプルプログラムのバージョン	Version 1.10
エミュレータ	E2OB (E2 オンボードエミュレータ)
使用ボード	Renesas Envision KIT RPBRX65N (製品型名: RTK5RX65N2CxxxxxBR)
ボード設定 (ジャンパ / スイッチ)	<SW1> Pin 1: ON Pin 2: OFF (デバッグを使用する) <SW4> Pin 1: OFF Pin 2: don't care (デバッグを使用する) <その他> デフォルト

### 3. 関連ドキュメント

本サンプルに関連するドキュメントを以下に示します。併せて参照してください。

Firmware Integration Technology ユーザーズマニュアル (R01AN1833)  
RX ファミリ ボードサポートパッケージモジュール Firmware Integration Technology (R01AN1685)  
RX ファミリ グラフィック LCD コントローラモジュール Firmware Integration Technology (R01AN3609)  
Renesas e<sup>2</sup> studio スマート・コンフィグレータ ユーザーガイド (R20AN0451)  
RX65N グループ Renesas Starter Kit+ for RX65N-2MB ユーザーズマニュアル (R20UT3888)  
RX65N Group RX65N Envision Kit User's Manual (R01UH0761)  
RX65N グループ、RX651 グループ ユーザーズマニュアル ハードウェア編 (R01UH0590)  
RX72N グループ Renesas Starter Kit+ for RX72N ユーザーズマニュアル (R20UT4436)  
RX72N Group RX72N Envision Kit User's Manual (R20UT4788)  
RX72N グループ ユーザーズマニュアル ハードウェア編 (R01UH0824)

最新版がある場合、最新版に差し替えて使用してください。最新版はルネサスエレクトロニクスホームページで確認および入手してください。

#### 4. サンプルプロジェクトの構成

本サンプルのプロジェクト構成を示します。なお、FIT モジュールや統合開発環境で自動生成されるファイルの詳細は除きます。

サンプルプロジェクトはボードごとに4つ用意しています。プロジェクト構成自体は同じで、QE for Display[RX]を使い LCD に合わせて出力したヘッダファイルとマイコンに依存するファイルが異なります。図 4-1 プロジェクトのフォルダ構成 では、代表して RSK RX72N プロジェクトのフォルダ構成を詳細に示しています。

表 4-1 サンプルのプロジェクト

プロジェクト名	概要
QE_for_Display_sample_RX72N_RSK	RSK RX72N で動作するプロジェクト
QE_for_Display_sample_RX72N_Envision	Envision RX72N で動作するプロジェクト
QE_for_Display_sample_RX65N_RSK	RSK RX65N で動作するプロジェクト
QE_for_Display_sample_RX65N_Envision	Envision RX65N で動作するプロジェクト

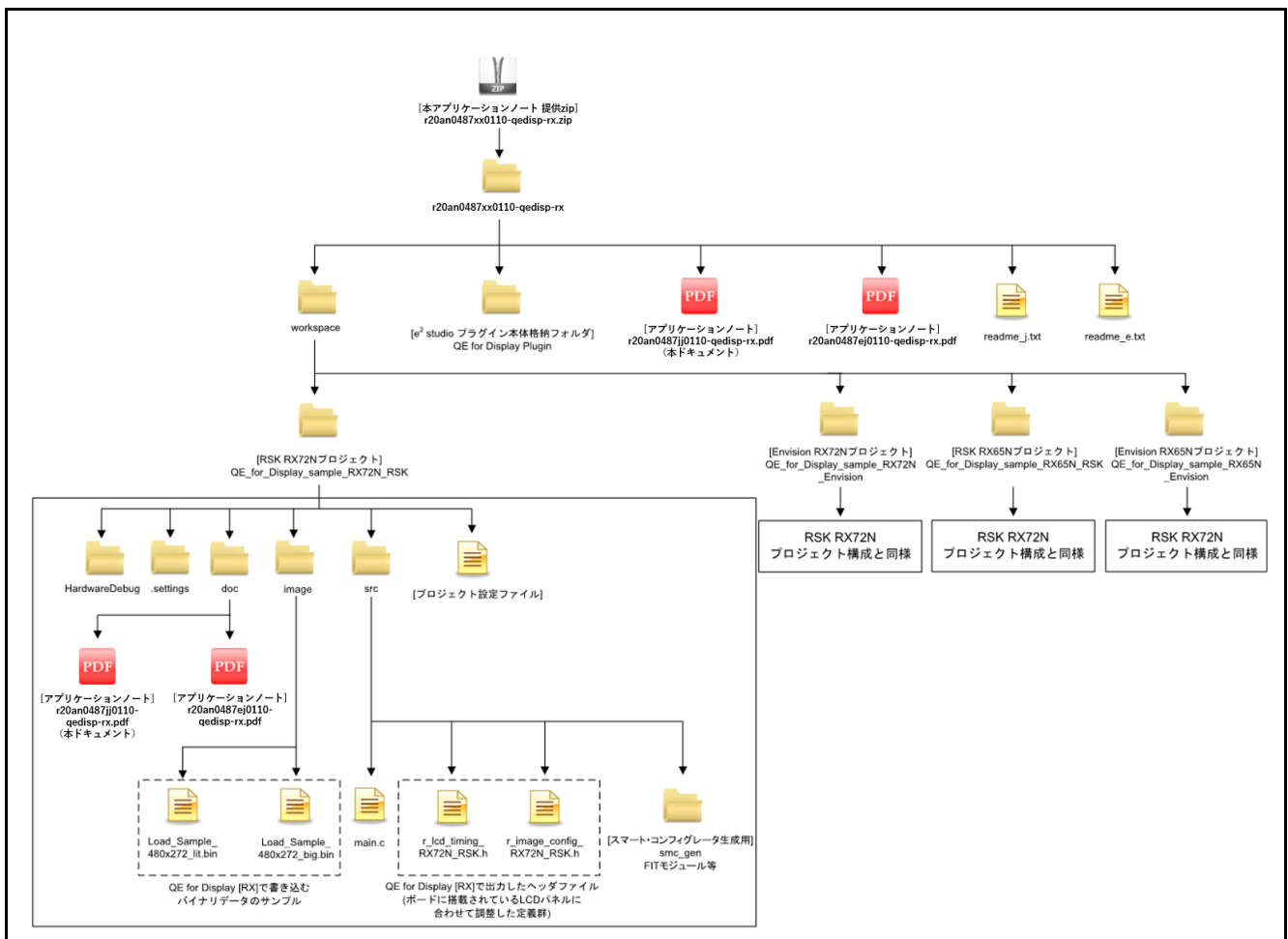


図 4-1 プロジェクトのフォルダ構成

## 5. サンプルプロジェクトの実行手順

本章では本サンプルプロジェクトで QE for Display [RX] のリアルタイム調整を行うまでの実行手順を示します。QE for Display [RX] のリアルタイム調整機能を使用するには、GLCDC の初期化が完了している必要があります。GLCDC の初期化はユーザプログラムで実行します。

本サンプルのプログラムでは GLCDC の初期化を `glcdc_initialize` 関数で行っています。`glcdc_initialize` 関数の実行が完了してから QE for Display [RX] によるリアルタイム調整を開始してください。本サンプルの詳細については、7. ソフトウェア説明を参照してください。

なお、本プロジェクトを動作させる前に 2. 動作環境に示すジャンパ設定がある場合は、必ず設定してください。

QE for Display [RX] の使用方法については、8. QE for Display [RX] の使用方法を参照してください。

### 事前準備

1. QE for Display [RX] のインストール

### 実行手順

2. プロジェクトのインポート
3. プロジェクトのビルド
4. デバッガ接続とプログラムの実行
5. QE for Display [RX] によるリアルタイム調整

## 5.1 QE for Display [RX]のインストール

まず初めに QE for Display [RX]を統合開発環境 e<sup>2</sup> studio にインストールします。インストールは下記の手順で行います。

### インストール方法

1. e<sup>2</sup> studio を起動する。
2. [ヘルプ(H)]→[新規ソフトウェアのインストール...]メニューを選択し、[インストール]ダイアログを開く。
3. [追加(A)...]ボタンを押下し、[リポジトリを追加]ダイアログを開く。
4. [アーカイブ(A)...]ボタンを押下し、開いたファイル選択ダイアログで、インストール用 zip ファイル(RenesasQE\_display\_package\_V130.zip)を選択し、[開く(O)]ボタンを押下する。
5. [リポジトリを追加]ダイアログで、[OK]ボタンを押下する。
6. [必要なソフトウェアを見つけるために、インストール中に更新サイト全てに接続(C)]チェックボックスを外します。
7. [インストール]ダイアログに、表示された[Renesas QE for Display[RX]]チェックボックスをチェックし、[次へ(N)>]ボタンを押下する。
8. インストール対象が [Renesas QE for Display[RX]] となっていることを確認し、[次へ(N)>]ボタンを押下する。
9. ライセンスを確認した後、[使用条件の条項に同意します(A)]ラジオ・ボタンを選択し、[終了(F)]ボタンを押下する。
10. 信頼する証明書を選択ダイアログが表示された場合、表示された証明書をチェックした後、[OK]ボタンを押下してインストールを継続する。
11. e<sup>2</sup> studio の再起動を促されるので再起動を行う。

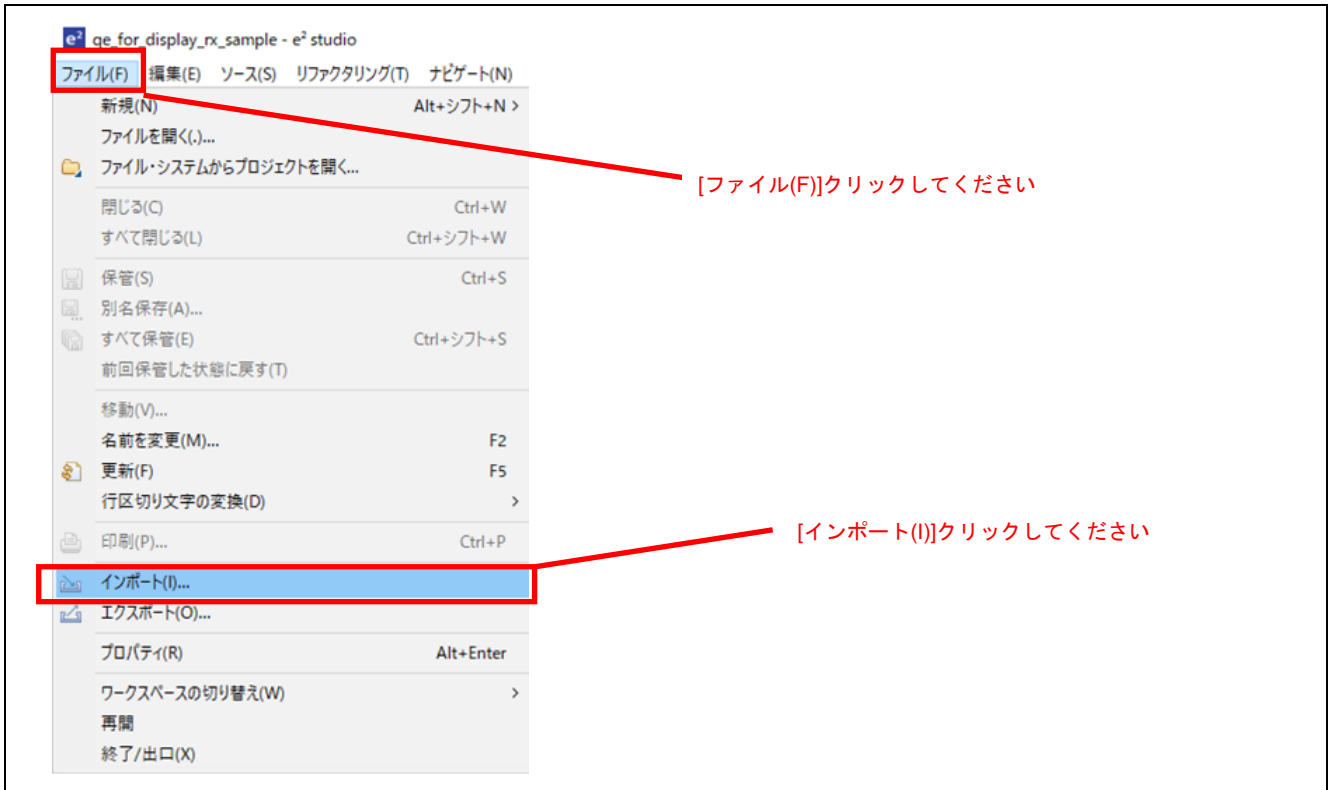
尚、アンインストールは下記の手順で行います。

### アンインストール方法

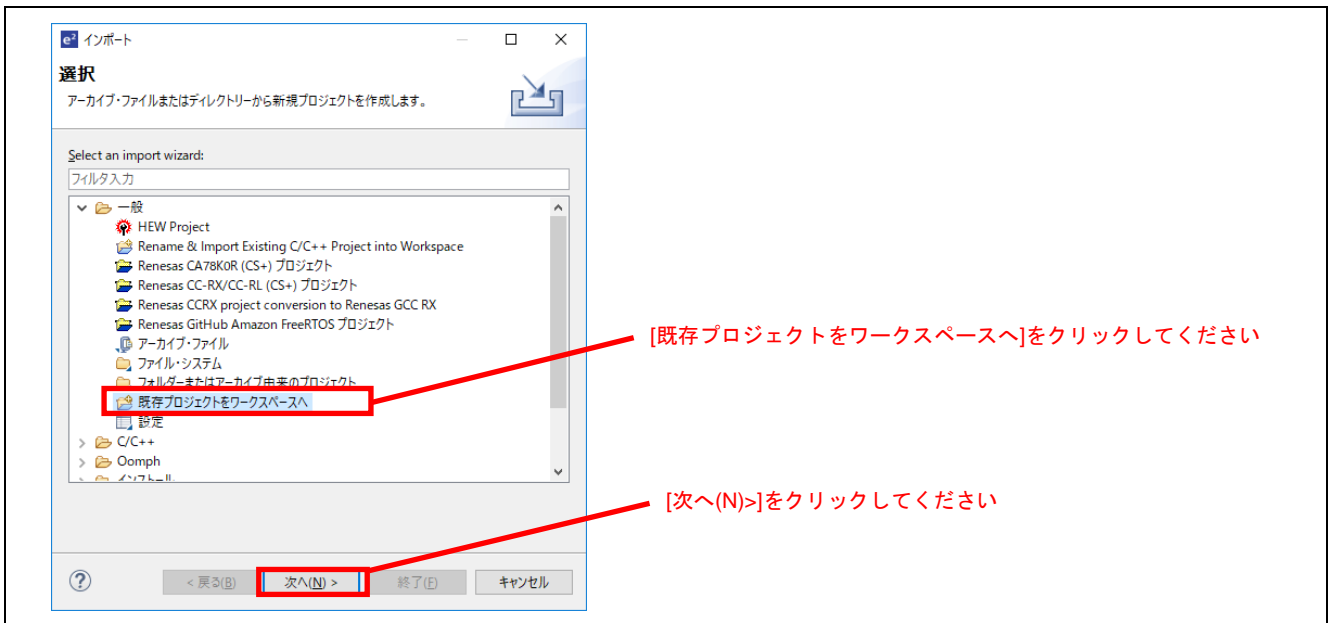
1. e<sup>2</sup> studio を起動する。
2. [ヘルプ(H)]→[e2studio について(A)]メニューを選択し、[インストール詳細(i)]ボタンを押下して、[e<sup>2</sup> studio のインストール詳細]ダイアログを開く。
3. [インストールされたソフトウェア]タブに表示されている[Renesas QE for Display[RX]]を選択し、[アンインストール(U)...]ボタンを押下して、[アンインストール]ダイアログを開く。
4. 表示された内容を確認し、[終了(F)]ボタンを押下する。
5. e<sup>2</sup> studio の再起動を促されるので再起動を行う。

## 5.2 プロジェクトのインポート

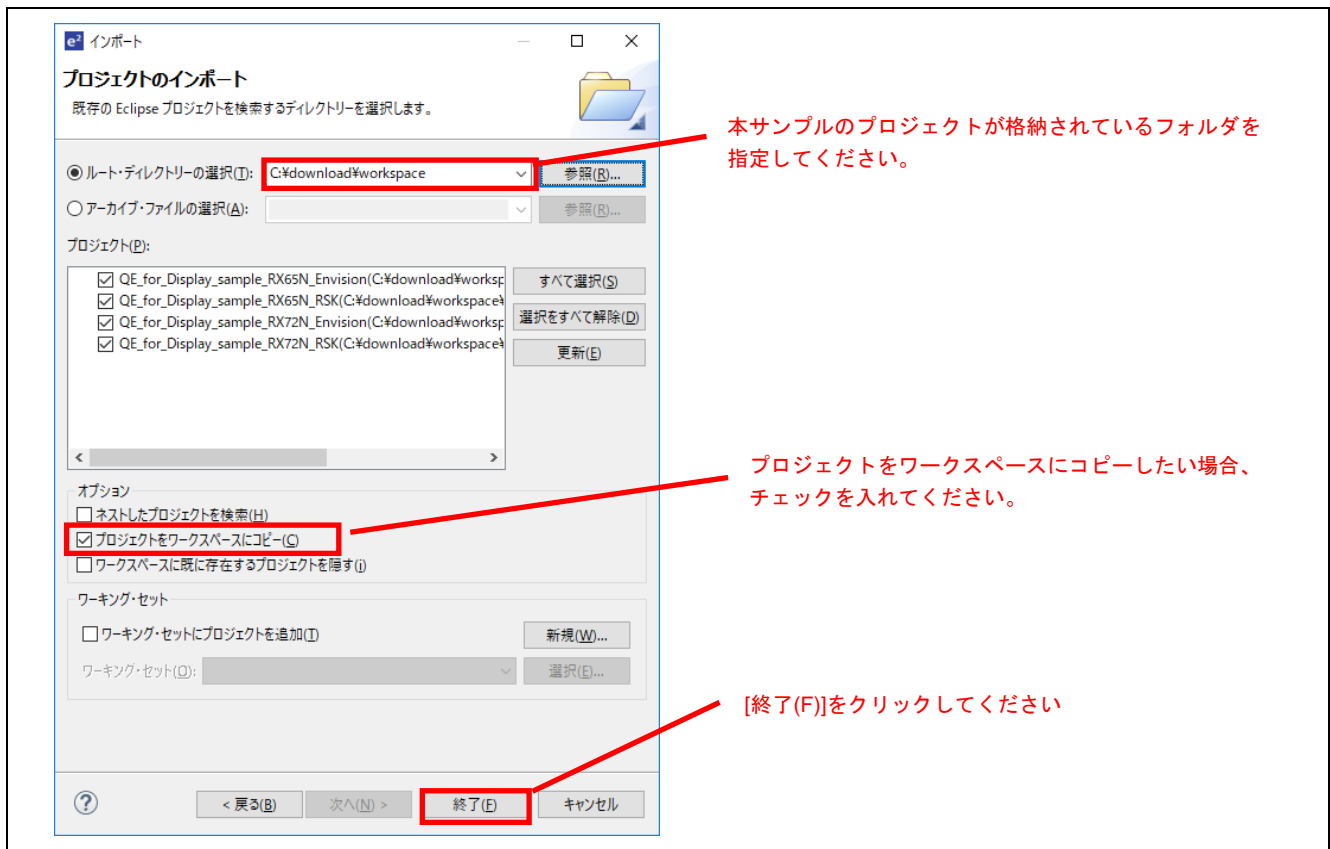
1. [ファイル(F)]をクリックしてください。
2. [インポート(I)]をクリックしてください。



3. [一般]の [既存プロジェクトをワークスペースへ]をクリックしてください。
4. [次へ(N)>]をクリックしてください。



- [ルート・ディレクトリーの選択(T):]のコンボボックスに本サンプルのプロジェクトが格納されているフォルダを指定してください。
- [終了(F)]をクリックしてください。



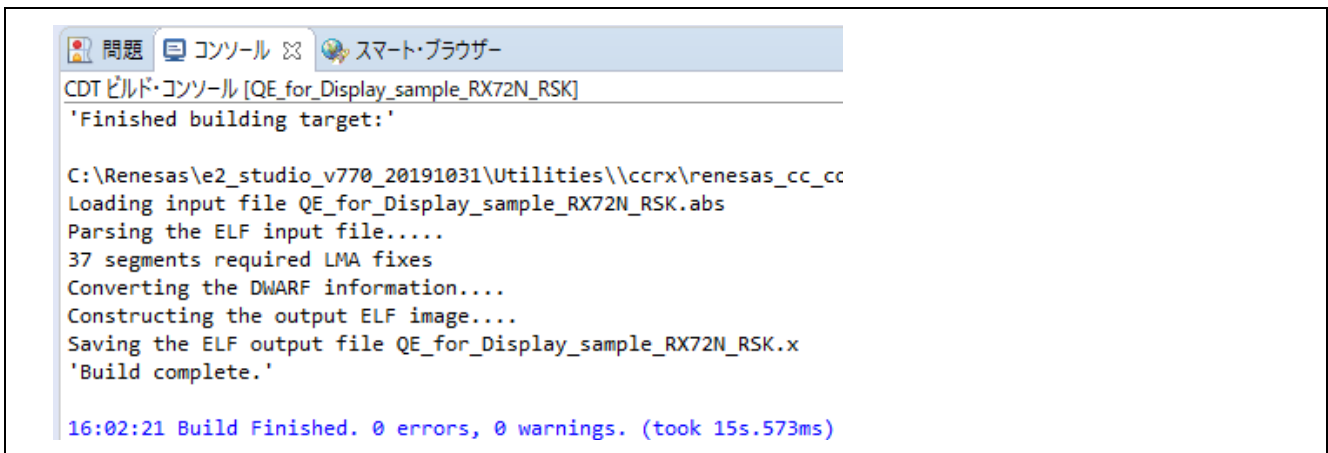
### 5.3 プロジェクトのビルド

以下の手順に従い、プロジェクトをビルドしてロードモジュールを作成してください。

1. ビルドするプロジェクト（例：QE\_for\_Display\_sample\_RX72N\_RSK HardwareDebug）をクリックしてください。
2. [Build]をクリックしてください。



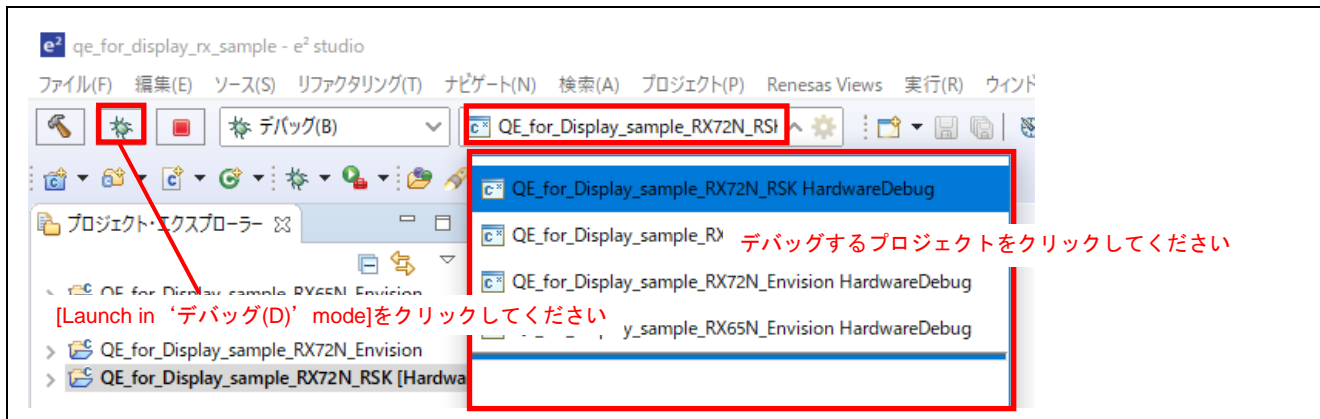
3. 「コンソール」パネルに「Build complete.」と表示されたらビルド完了です。



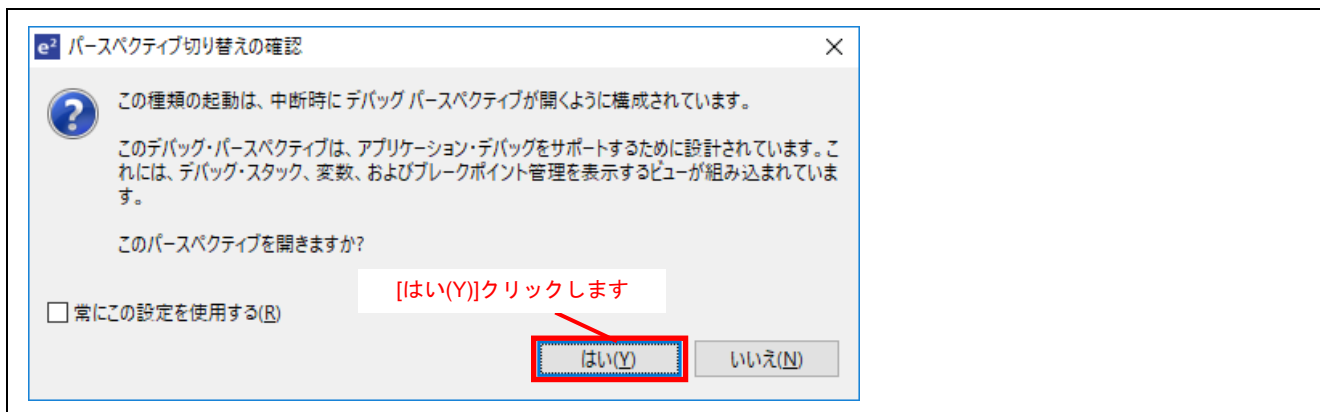


## 5.4 デバッグ接続とプログラムの実行

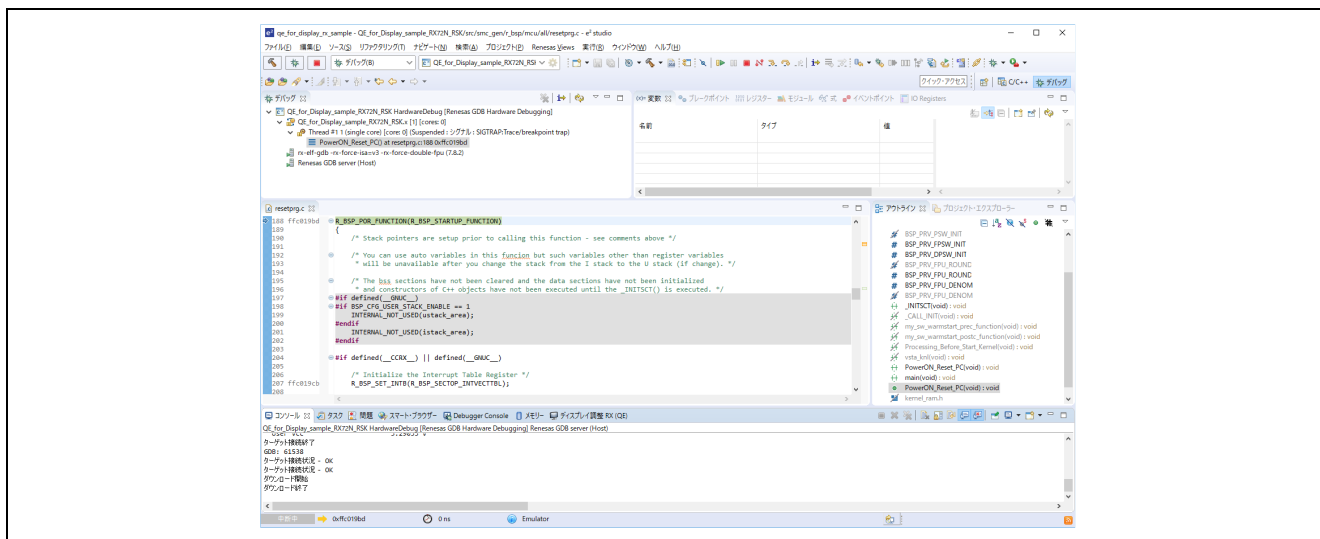
1. デバッグするプロジェクト（例：QE\_for\_Display\_sample\_RX72N\_RSK HardwareDebug）をクリックしてください。
2. [Launch in 'デバッグ(D)' mode]をクリックしてください。



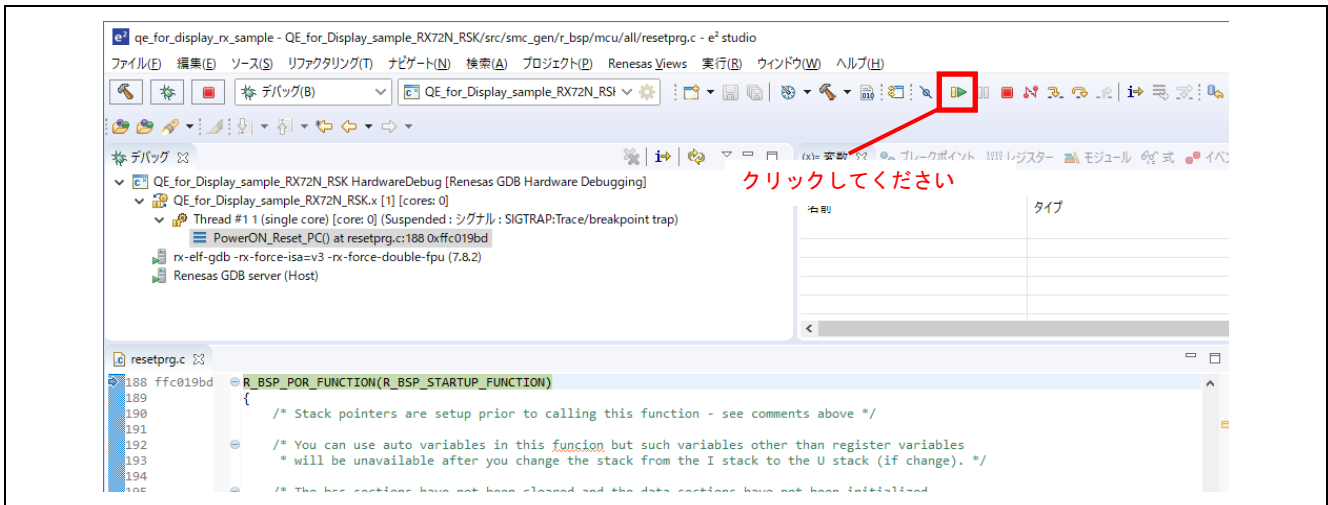
3. 以下のメッセージが表示されたら、[はい(Y)]をクリックしてください。



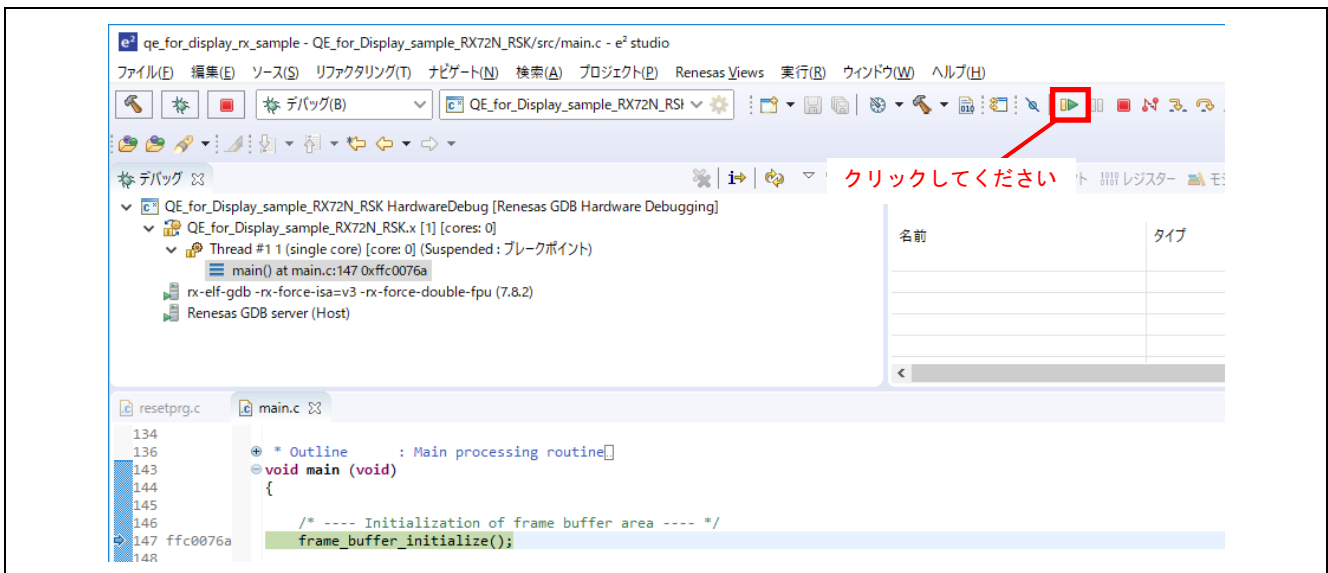
4. ロードモジュールのダウンロードが完了すると、[デバッグ]パースペクティブが開きます。



5. ツールバーの[再開]をクリックしてください。プログラムが実行され、main 関数の先頭でブレークします。



6. main 関数の先頭でブレークした後に、もう一度ツールバーの[再開]をクリックしてください。



7. 表示機器の設定が正しく行われた場合、LCD パネルに以下の画面が表示されます。



## 5.5 QE for Display [RX]によるリアルタイム調整

1. LCDパネルに画面が表示されたら、QE for Display [RX] を起動し、リアルタイム調整を開始してください。

The screenshot shows the Renesas Studio IDE with the QE for Display [RX] configuration window open. The window displays the GLCDC block diagram, including components like Background generator, Image Synthesizer, Blending, Output Controller, and Buffer Area. The Output Controller is configured with options like Brightness, Contrast, Gamma Correction, and Image Processing. The main window shows the source code for the main.c file, which includes initialization functions and a while loop with a nop() call.

```
133
134
135  * Outline : Main processing routine
136  void main (void)
137  {
138
139    /* ---- Initialization of frame buffer area ---- */
140    frame_buffer_initialize();
141
142    /* ---- Initialization of the GLCDC, and start display ---- */
143    glcdc_initialize();
144
145    while (1)
146    {
147      nop();
148    }
149  } /* End of function main() */
```

## 6. ハードウェア説明

### 6.1 ハードウェア構成

本サンプルで使用する LCD パネルを表 6-1 に示します。

表 6-1 サンプルで使用する LCD パネル

ボード	LCD パネル製品情報
RSK RX72N RSK RX65N	メーカー: Newhaven Display 社製 型番: NHD-4.3-480272EF-ATXL#-CTP 画面サイズ: 480×272 同期信号: VS、HS、DE (3 信号) タッチコントローラ搭載 (本サンプルでは使用しません)
Envision RX72N Envision RX65N	メーカー: EastRising 社製 型番: ER-TFT043-3 画面サイズ: 480×272 同期信号: VS、HS、DE (3 信号) タッチコントローラ搭載 (本サンプルでは使用しません)

### 6.2 端子機能

各 RSK および Envision で使用する端子と機能を以下に示します。使用する製品に合わせて端子を選択してください。なお、端子機能は Smart Configurator を使用して設定できます。

表 6-2 使用端子と機能 (RSK RX72N)

接続デバイス	端子名	入出力	内容
NHD-4.3-480272EF-ATXL#-CTP	P14/LCD_CLK-B	出力	パネルクロック出力
	P13/LCD_TCON 0-B	出力	同期信号(VSYNC)出力
	PJ2/LCD_TCON 2-B	出力	同期信号(HSYNC)出力
	PB1/LCD_TCON 3-B	出力	同期信号(DE)出力
	PC5/LCD_DATA 0-B	出力	LCD 信号出力 R[3]
	P82/LCD_DATA 1-B	出力	LCD 信号出力 R[4]
	P81/LCD_DATA 2-B	出力	LCD 信号出力 R[5]
	P80/LCD_DATA 3-B	出力	LCD 信号出力 R[6]
	PC4/LCD_DATA 4-B	出力	LCD 信号出力 R[7]
	P55/LCD_DATA 5-B	出力	LCD 信号出力 G[2]
	P54/LCD_DATA 6-B	出力	LCD 信号出力 G[3]
	P11/LCD_DATA 7-B	出力	LCD 信号出力 G[4]
	P83/LCD_DATA 8-B	出力	LCD 信号出力 G[5]
	PC7/LCD_DATA 9-B	出力	LCD 信号出力 G[6]
	PC6/LCD_DATA 10-B	出力	LCD 信号出力 G[7]
	PJ0/LCD_DATA 11-B	出力	LCD 信号出力 B[3]
	P85/LCD_DATA 12-B	出力	LCD 信号出力 B[4]
P84/LCD_DATA 13-B	出力	LCD 信号出力 B[5]	
P57/LCD_DATA 14-B	出力	LCD 信号出力 B[6]	

接続デバイス	端子名	入出力	内容
	P56/LCD_DATA 15-B	出力	LCD 信号出力 B[7]
	P27/汎用入出力ポート	出力	バックライト (プログラムで制御)
	PK4/汎用入出力ポート	出力	パネルリセット (プログラムで制御)

表 6-3 使用端子と機能 (Envision RX72N)

接続デバイス	端子名	入出力	内容
ER-TFT043-3	PB5/LCD_CLK-B	出力	パネルクロック出力
	PB4/LCD_TCON 0-B	出力	同期信号(VSYNC)出力
	PB2/LCD_TCON 2-B	出力	同期信号(HSYNC)出力
	PB1/LCD_TCON 3-B	出力	同期信号(DE)出力
	PB0/LCD_DATA 0-B	出力	LCD 信号出力 B[3]
	PA7/LCD_DATA 1-B	出力	LCD 信号出力 B[4]
	PA6/LCD_DATA 2-B	出力	LCD 信号出力 B[5]
	PA5/LCD_DATA 3-B	出力	LCD 信号出力 B[6]
	PA4/LCD_DATA 4-B	出力	LCD 信号出力 B[7]
	PA3/LCD_DATA 5-B	出力	LCD 信号出力 G[2]
	PA2/LCD_DATA 6-B	出力	LCD 信号出力 G[3]
	PA1/LCD_DATA 7-B	出力	LCD 信号出力 G[4]
	PA0/LCD_DATA 8-B	出力	LCD 信号出力 G[5]
	PE7/LCD_DATA 9-B	出力	LCD 信号出力 G[6]
	PE6/LCD_DATA 10-B	出力	LCD 信号出力 G[7]
	PE5/LCD_DATA 11-B	出力	LCD 信号出力 R[3]
	PE4/LCD_DATA 12-B	出力	LCD 信号出力 R[4]
	PE3/LCD_DATA 13-B	出力	LCD 信号出力 R[5]
	PE2/LCD_DATA 14-B	出力	LCD 信号出力 R[6]
	PE1/LCD_DATA 15-B	出力	LCD 信号出力 R[7]
	P67/汎用入出力ポート	出力	バックライト (プログラムで制御)
	PB3/汎用入出力ポート	出力	パネルリセット (プログラムで制御)

表 6-4 使用端子と機能 (RSK RX65N)

接続デバイス	端子名	入出力	内容
NHD-4.3-480272EF-ATXL#-CTP	PB5/LCD_CLK-B	出力	パネルクロック出力
	PB4/LCD_TCON 0-B	出力	同期信号(VSYNC)出力
	PB2/LCD_TCON 2-B	出力	同期信号(HSYNC)出力
	PB1/LCD_TCON 3-B	出力	同期信号(DE)出力
	PB0/LCD_DATA 0-B	出力	LCD 信号出力 R[3]
	PA7/LCD_DATA 1-B	出力	LCD 信号出力 R[4]
	PA6/LCD_DATA 2-B	出力	LCD 信号出力 R[5]
	PA5/LCD_DATA 3-B	出力	LCD 信号出力 R[6]
	PA4/LCD_DATA 4-B	出力	LCD 信号出力 R[7]
	PA3/LCD_DATA 5-B	出力	LCD 信号出力 G[2]
	PA2/LCD_DATA 6-B	出力	LCD 信号出力 G[3]
	PA1/LCD_DATA 7-B	出力	LCD 信号出力 G[4]

接続デバイス	端子名	入出力	内容
	PA0/LCD_DATA 8-B	出力	LCD 信号出力 G[5]
	PE7/LCD_DATA 9-B	出力	LCD 信号出力 G[6]
	PE6/LCD_DATA 10-B	出力	LCD 信号出力 G[7]
	PE5/LCD_DATA 11-B	出力	LCD 信号出力 B[3]
	PE4/LCD_DATA 12-B	出力	LCD 信号出力 B[4]
	PE3/LCD_DATA 13-B	出力	LCD 信号出力 B[5]
	PE2/LCD_DATA 14-B	出力	LCD 信号出力 B[6]
	PE1/LCD_DATA 15-B	出力	LCD 信号出力 B[7]
	PB7/汎用入出力ポート	出力	バックライト（プログラムで制御）
	P97/汎用入出力ポート	出力	パネルリセット（プログラムで制御）

表 6-5 使用端子と機能（Envision RX65N）

接続デバイス	端子名	入出力	内容
ER-TFT043-3	PB5/LCD_CLK-B	出力	パネルクロック出力
	PB4/LCD_TCON 0-B	出力	同期信号(VSYNC)出力
	PB2/LCD_TCON 2-B	出力	同期信号(HSYNC)出力
	PB1/LCD_TCON 3-B	出力	同期信号(DE)出力
	PB0/LCD_DATA 0-B	出力	LCD 信号出力 B[3]
	PA7/LCD_DATA 1-B	出力	LCD 信号出力 B[4]
	PA6/LCD_DATA 2-B	出力	LCD 信号出力 B[5]
	PA5/LCD_DATA 3-B	出力	LCD 信号出力 B[6]
	PA4/LCD_DATA 4-B	出力	LCD 信号出力 B[7]
	PA3/LCD_DATA 5-B	出力	LCD 信号出力 G[2]
	PA2/LCD_DATA 6-B	出力	LCD 信号出力 G[3]
	PA1/LCD_DATA 7-B	出力	LCD 信号出力 G[4]
	PA0/LCD_DATA 8-B	出力	LCD 信号出力 G[5]
	PE7/LCD_DATA 9-B	出力	LCD 信号出力 G[6]
	PE6/LCD_DATA 10-B	出力	LCD 信号出力 G[7]
	PE5/LCD_DATA 11-B	出力	LCD 信号出力 R[3]
	PE4/LCD_DATA 12-B	出力	LCD 信号出力 R[4]
	PE3/LCD_DATA 13-B	出力	LCD 信号出力 R[5]
	PE2/LCD_DATA 14-B	出力	LCD 信号出力 R[6]
	PE1/LCD_DATA 15-B	出力	LCD 信号出力 R[7]
	P66/汎用入出力ポート	出力	バックライト（プログラムで制御）
	P63/汎用入出力ポート	出力	パネルリセット（プログラムで制御）

## 7. ソフトウェア説明

### 7.1 動作概要

本サンプルは、クロックや割り込みなどのCPUの初期化を行い、QE for Display [RX]から出力されたヘッダファイルをもとに GLCDC の動作設定を行います。

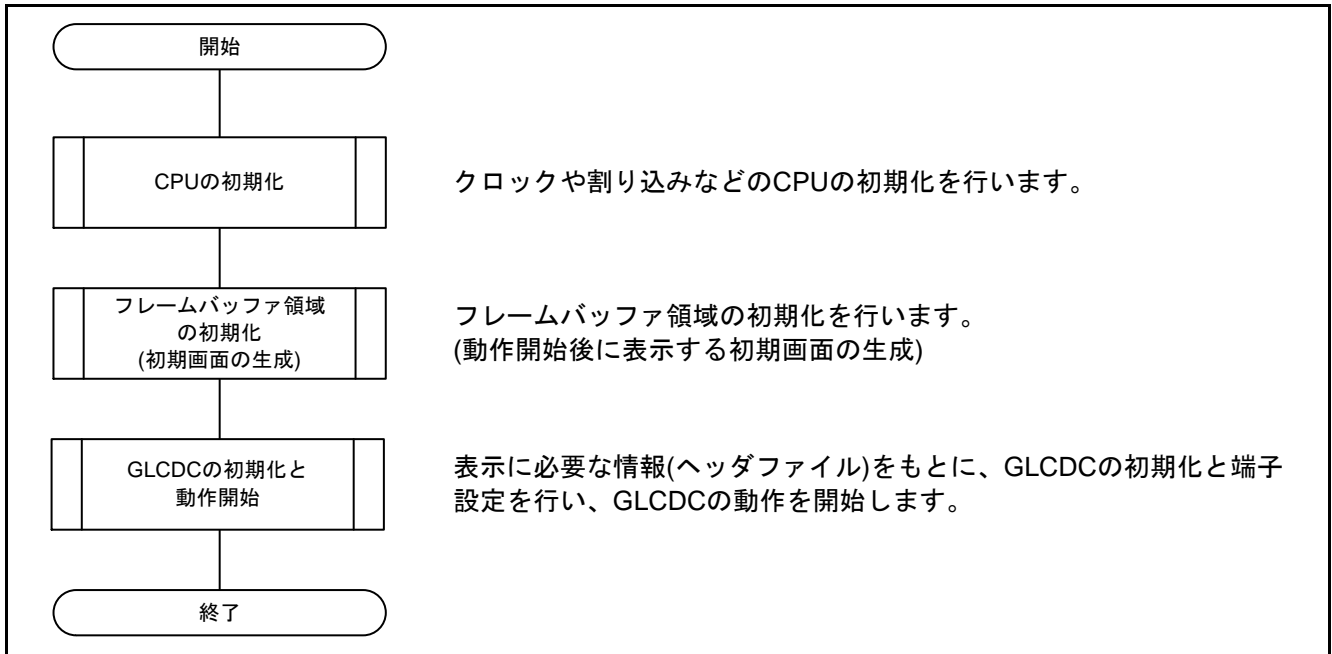


図 7-1 サンプルプログラムの動作概要

## 7.2 GLCDC の動作設定詳細

図 7-2 に GLCDC の動作設定詳細を示します。GLCDC の設定は、GLCDC FIT モジュールを通じて行います。GLCDC FIT モジュールの初期化関数である R\_GLCDC\_Open 関数の引数には、GLCDC FIT モジュールが用意している glcdc\_cfg\_t 構造体の変数を設定します。この構造体変数に、画面表示のためのいくつかのパラメータを設定します。

このパラメータに設定する情報は、以下に分類されます。

- |                                  |                              |
|----------------------------------|------------------------------|
| a. GLCDC が出力する同期信号や RGB 信号に関する設定 | → QE for Display [RX] を使って編集 |
| b. 入力データに対する出力補正の設定              | → QE for Display [RX] を使って編集 |
| c. GLCDC に入力する画像データに関する設定        | → ユーザがプログラムを直接編集             |
| d. 割り込み関連の設定                     | → ユーザがプログラムを直接編集             |

QE for Display [RX]では、主に LCD パネルの仕様に依存する a. と b. のパラメータ設定をサポートしています。残りの c. と d. に関しては、使用する画像の形式やシステムに応じてユーザがプログラムを直接編集する必要があります。ユーザは構造体変数の c. d. に該当するパラメータに対して値を指定します。

パラメータの設定が完了してから、GLCDC FIT モジュールの初期化関数 (R\_GLCDC\_Open) にパラメータを渡します。この後、端子の設定を行います。

- |             |                             |
|-------------|-----------------------------|
| GLCDC の端子設定 | → Smart Configurator を使って編集 |
| ボードの端子設定    | → ユーザがプログラムを直接編集            |

端子設定後、GLCDC FIT モジュールの制御関数 (R\_GLCDC\_Control) で画面表示を開始します。

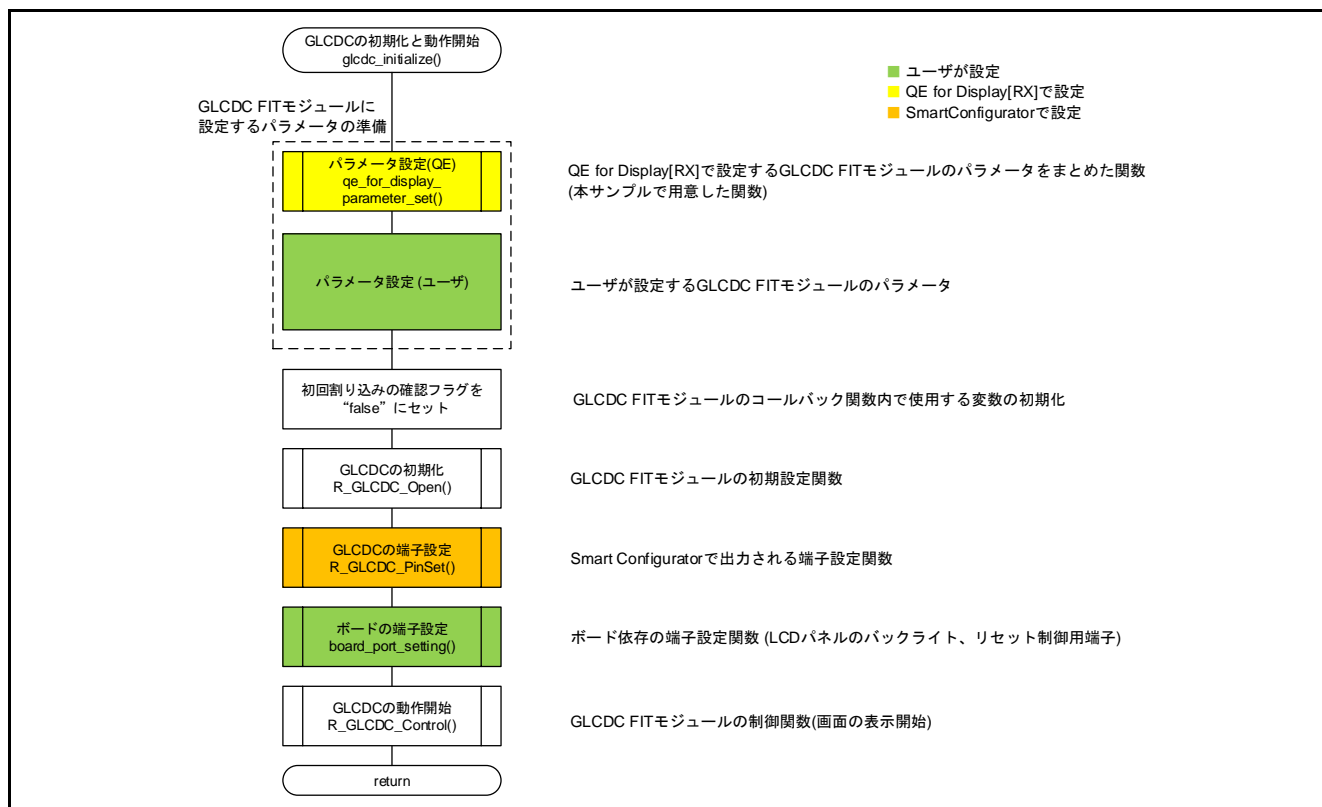


図 7-2 GLCDC の動作設定詳細



### 7.2.1 パラメータ設定(QE)

ここで呼び出す `qe_for_display_parameter_set` 関数は、本サンプルで用意した関数です。GLCDC FIT モジュールに設定するパラメータの内、QE for Display [RX]が出力する define 定義を使うパラメータをまとめています。

### 7.2.2 パラメータ設定(ユーザ)

GLCDC FIT モジュールに設定するパラメータの内、QE for Display [RX]で調整しないパラメータを設定します。使用する画像の形式やシステムに応じてユーザがプログラムを直接編集する必要があります。ユーザは構造体変数の `c.d.` に該当するパラメータに対して値を設定します。本サンプルにおける設定値は、7.3 GLCDC FIT モジュールのパラメータと QE for Display [RX]が出力するヘッダとの対応を参照してください。

### 7.2.3 GLCDC の端子設定

6.2 端子機能で GLCDC の端子設定は Smart Configurator で設定できると説明しました。ここで呼び出す `R_GLCDC_PinSet` 関数は、Smart Configurator が生成した `r_glcdc_rx_pinset.c` ファイル内に実装されています。

Smart Configurator で GLCDC の端子設定ができる様子を図 7-3 に示します。LCD パネルの仕様と接続に応じて、各端子機能を使用する/使用しないの選択と、各端子機能に割り当てるポート番号を指定します。端子設定の方法については、Smart Configurator のマニュアルを参照してください。

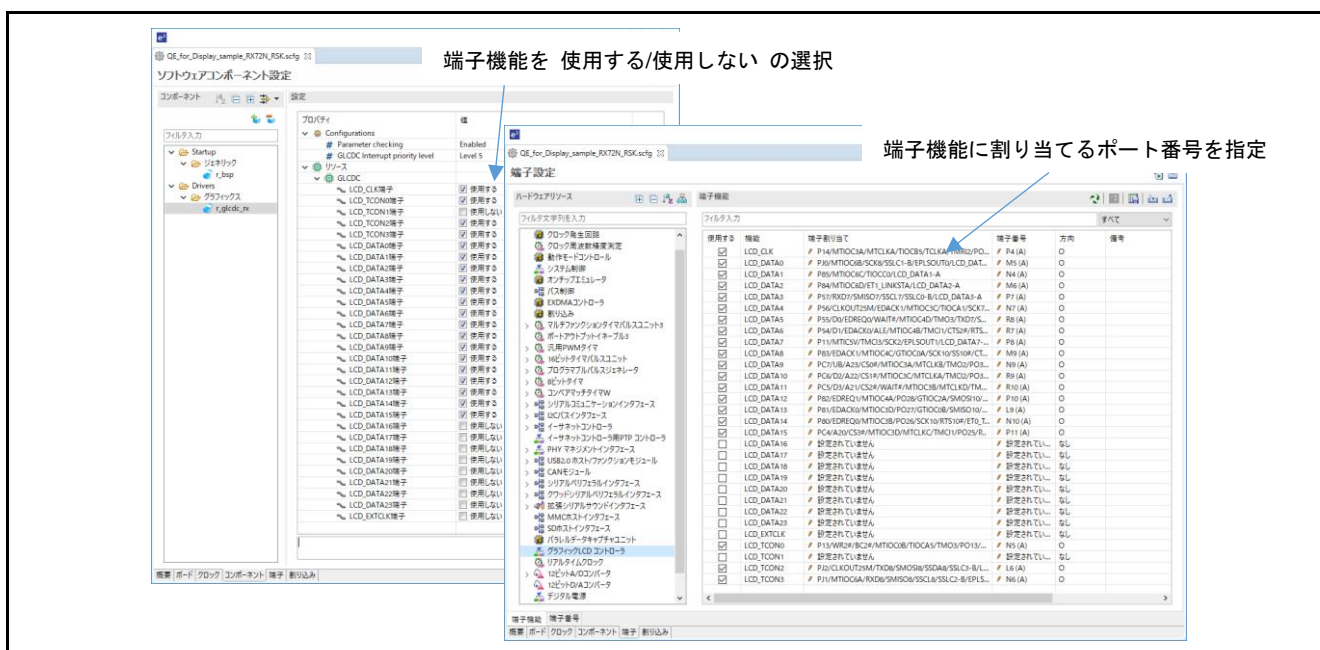


図 7-3 GLCDC の端子設定画面 (Smart Configurator)

### 7.2.4 ボードの端子設定

各ボードには、GLCDC が制御する端子の他に LCD パネルのバックライトやリセット端子を制御するためのポートが接続されています。

ここで呼び出す `board_port_setting` 関数では、LCD パネルのバックライトやリセット端子の制御を行います。これら端子の制御はユーザがプログラムを直接編集する必要があります。

### 7.3 GLCDC FIT モジュールのパラメータと QE for Display [RX]が出力するヘッダとの対応

GLCDC FIT モジュールに設定するパラメータ(R\_GLCDC\_Open 関数の引数に設定する glcdc\_cfg\_t 構造体メンバ)と、QE for Display [RX]が出力する define 定義の対応を表 7-1 に示します。サンプルの設定値の列で、“←”の記載になっているものは、QE for Display [RX] が出力する define 定義で設定しています。“←”以外の記載になっているものは、ユーザが設定します。パラメータの詳細は、GLCDC FIT モジュールのマニュアルを参照してください。

表 7-1 GLCDC FIT モジュールのパラメータと QE for Display [RX] が出力する定義の対応

概略	構造体メンバ	QE for Display [RX]が出力する define 定義	サンプルの設定値	設定内容
水平バックポーチ	output.htiming.back_porch	LCD_CH0_W_HBP	←	水平バックポーチ期間
水平アサート幅	output.htiming.sync_width	LCD_CH0_W_HSYNC	←	水平同期信号期間
垂直バックポーチ	output.vtiming.back_porch	LCD_CH0_W_VBP	←	垂直バックポーチ期間
垂直アサート幅	output.vtiming.sync_width	LCD_CH0_W_VSYNC	←	垂直同期信号期間
水平有効表示幅	output.htiming.display_cyc	LCD_CH0_DISP_HW	←	水平有効表示期間
垂直有効表示幅	output.vtiming.display_cyc	LCD_CH0_DISP_VW	←	垂直有効表示期間
水平フロントポーチ	output.htiming.front_porch	LCD_CH0_W_HFP	←	水平フロントポーチ期間
垂直フロントポーチ	output.vtiming.front_porch	LCD_CH0_W_VFP	←	垂直フロントポーチ期間
コールバック関数へのポインタ	p_callback	-	glcdc_callback	割り込み要因発生時にポインタが示すアドレスのコールバック関数を実行します
クロックソース	output.clksrc	-	GLCDC_CLK_SRC_INTERNAL	PLL クロックを使用
クロックの分周比	output.clock_div_ratio	-	GLCDC_PANEL_CLK_DIVISOR_24	LCD_CLK の分周比の設定
出力データフォーマット	output.format	LCD_CH0_OUT_FORMAT	←	出力データフォーマット
TCON、DATA の出力位相制御	output.sync_edge	LCD_CH0_OUT_EDGE	←	LCD_CLK の立ち上がり/立ち下がりに同期して出力
水平同期信号(HSYNC)の出力端子	output.tcon_hsync	LCD_CH0_TCON_PIN_HSYNC	←	HSYNC の出力に使用する TCON を選択
水平同期信号(HSYNC)の極性	output.hsync_polarity	LCD_CH0_TCON_POL_HSYNC	←	極性をロー/ハイアクティブに設定

概略	構造体メンバ	QE for Display [RX]が出力する define 定義	サンプルの設定値	設定内容
垂直同期信号(VSYNC)の出力端子	output.tcon_vsync	LCD_CH0_TCON_PIN_VSYNC	←	VSYNC の出力に使用する TCON を選択
垂直同期信号(VSYNC)の極性	output.vsync_polarity	LCD_CH0_TCON_POL_VSYNC	←	極性をロー/ハイアクティブに設定
データイネーブル信号(DE)の出力端子	output.tcon_de	LCD_CH0_TCON_PIN_DE	←	DE の出力に使用する TCON を選択
データイネーブル信号(DE)の極性	output.data_enable_polarity	LCD_CH0_TCON_POL_DE	←	極性をロー/ハイアクティブに設定
背景色 R 値	output.bg_color.byte.r	-	0xCC	背景色の R 値の設定
背景色 G 値	output.bg_color.byte.g	-	0xCC	背景色の G 値の設定
背景色 B 値	output.bg_color.byte.b	-	0xCC	背景色の B 値の設定
フレームバッファの画像フォーマット	input.format	-	グラフィック 2: GLCDC_IN_FORMAT_16BITS_RGB565  グラフィック 1:未設定	カラーフォーマットを選択
フレームバッファの先頭アドレス	input.p_base	-	グラフィック 2: FRAME_BUFFER_BASE_ADDRESS(0x0080000)  グラフィック 1:NULL	フレームバッファの先頭アドレスの設定。 NULL を設定すると対象グラフィックを無効に設定 (glcdc_cfg_t.input 以下の構造体メンバの設定値は無視されます)
グラフィック 1,2 の背景色 R 値	input.bg_color.byte.r	-	グラフィック 2: 0xCC  グラフィック 1:未設定	グラフィック 1,2 の背景色の R 値の設定
グラフィック 1,2 の背景色 G 値	input.bg_color.byte.g	-	グラフィック 2: 0xCC  グラフィック 1:未設定	グラフィック 1,2 の背景色の G 値の設定
グラフィック 1,2 の背景色 B 値	input.bg_color.byte.b	-	グラフィック 2: 0xCC	グラフィック 1,2 の背景色の B 値の設定

概略	構造体メンバ	QE for Display [RX]が出力する define 定義	サンプルの設定値	設定内容
			グラフィック 1:未設定	
画像データの横幅	input.hsize	-	グラフィック 2: IMAGE_WIDTH(480)  グラフィック 1:未設定	グラフィック 1,2 の画像の横幅の設定
画像のデータの高さ	input.vsize	-	グラフィック 2: IMAGE_HEIGHT(272)  グラフィック 1:未設定	グラフィック 1,2 の画像の高さの設定
マクロラインオフセット	input.offset	-	グラフィック 2: (IMAGE_WIDTH(480) * IMAGE_PIXEL_SIZE(2))  グラフィック 1:未設定	グラフィック 1,2 のマクロラインオフセットの設定
グラフィック領域の枠の表示	input.frame_edge	-	グラフィック 2:false  グラフィック 1:未設定	グラフィック領域枠を表示/非表示に設定
表示開始位置 x 座標	input.coordinate.x	-	グラフィック 2:0  グラフィック 1:未設定	グラフィック領域水平開始位置の設定
表示開始位置 y 座標	input.coordinate.y	-	グラフィック 2:0  グラフィック 1:未設定	グラフィック領域垂直開始位置の設定
ブレンド処理の制御設定	blend.blend_control	-	グラフィック 2: GLCDC_BLEND_CONTROL_NONE  グラフィック 1:未設定	アルファブレンド処理の設定
画像の表示設定	blend.visible	-	グラフィック 2: true	画像を表示/非表示に設定

概略	構造体メンバ	QE for Display [RX]が出力する define 定義	サンプルの設定値	設定内容
			グラフィック 1:未設定	
矩形アルファブレンド領域の枠の表示	blend.frame_edge	-	グラフィック 2:false  グラフィック 1:未設定	矩形アルファブレンド領域の枠を表示/非表示に設定
固定アルファ値	blend.fixed_blend_value	-	グラフィック 2:0  グラフィック 1:未設定	固定アルファ値の設定 (blend_control が GLCDC_BLEND_CONTROL_FIXED のときのみ有効)
アルファ値の加減値	blend.fade_speed	-	グラフィック 2:0  グラフィック 1:未設定	アルファ値の加減値の設定 (blend_control が GLCDC_BLEND_CONTROL_FADEIN もしくは GLCDC_BLEND_CONTROL_FADEOUT のときのみ有効)
ブレンド処理開始位置の x 座標	blend.start_coordinate.x	-	グラフィック 2:0  グラフィック 1:未設定	矩形アルファブレンド領域水平幅、矩形アルファブレンド水平開始位置を設定 (blend_control が GLCDC_BLEND_CONTROL_NONE のときは無効)
ブレンド処理終了位置の x 座標	blend.end_coordinate.x	-	グラフィック 2:IMAGE_WIDTH(480)  グラフィック 1:未設定	
ブレンド処理の開始位置の y 座標	blend.start_coordinate.y	-	グラフィック 2:0  グラフィック 1:未設定	矩形アルファブレンド領域垂直幅、矩形アルファブレンド垂直開始位置を設定 (blend_control が GLCDC_BLEND_CONTROL_NONE のときは無効)
ブレンド処理の終了位置の y 座標	blend.end_coordinate.y	-	グラフィック 2:IMAGE_HEIGHT(272)  グラフィック 1:未設定	
クロマキー処理の有効、無効	chromakey.enbale	-	グラフィック 2: false  グラフィック 1:未設定	クロマキー処理を有効/無効に設定
クロマキー処理対象 R 値	chromakey.before.byte.r	-	グラフィック 2:0xFF  グラフィック 1:未設定	クロマキー処理対象 R 値の設定 (chromakey.enbale が false のときは無効)

概略	構造体メンバ	QE for Display [RX]が出力する define 定義	サンプルの設定値	設定内容
クロマキー処理対象 G 値	chromakey.before.byte.g	-	グラフィック 2: 0xFF グラフィック 1:未設定	クロマキー処理対象 G 値の設定 (chromakey.enbale が false のときは無効)
クロマキー処理対象 B 値	chromakey.before.byte.b	-	グラフィック 2: 0xFF グラフィック 1:未設定	クロマキー処理対象 B 値の設定 (chromakey.enbale が false のときは無効)
クロマキー置き換え後 A 値	chromakey.after.byte.a	-	グラフィック 2: 0xFF グラフィック 1:未設定	クロマキー処理で置き換えた後の A 値を設定 (chromakey.enbale が false のときは無効)
クロマキー置き換え後 R 値	chromakey.after.byte.r	-	グラフィック 2: 0xFF グラフィック 1:未設定	クロマキー処理で置き換えた後の R 値を設定 (chromakey.enbale が false のときは無効)
クロマキー置き換え後 G 値	chromakey.after.byte.g	-	グラフィック 2: 0xFF グラフィック 1:未設定	クロマキー処理で置き換えた後の G 値を設定 (chromakey.enbale が false のときは無効)
クロマキー置き換え後 B 値	chromakey.after.byte.b	-	グラフィック 2: 0xFF グラフィック 1:未設定	クロマキー処理で置き換えた後の B 値を設定 (chromakey.enbale が false のときは無効)
出力データのビットエンディアン	output.endian	-	GLCDC_EN DIAN_LITTLE	リトルエンディアン/ビッグエンディアンに設定
出力データのピクセル順序	output.color_order	-	RSK RX65N: GLCDC_COLOR_ORDER_BGR  RSK RX72N, Envision RX72N, Envision RX65N: GLCDC_COLOR_ORDER_RGB	出力データのピクセル順序を R-G-B 順/ B-G-R 順に設定  ※ボードによって設定が異なります。
補正処理の実行順序	output.correction_proc_order	IMGC_OUTCTL_CALIB_ROUTE	←	輝度/コントラスト補正の後にガンマ補正を実行する。 または ガンマ補正の後に輝度/コントラスト補正を実行するように設定

概略	構造体メンバ	QE for Display [RX]が出力する define 定義	サンプルの設定値	設定内容
ディザ処理のモード選択	output.dithering.dithering_on	IMGC_DITHER_ACTIVE	←	True のとき 0 捨 1 入または 2x2 パターンディザに設定。 False のとき、切り捨てモードに設定 (glcdc_cfg_t.output.dithering 以下の構造体メンバの設定値は無視されます)
ディザ処理のモード選択 2	output.dithering.dithering_mode	IMGC_DITHER_MODE	←	0 捨 1 入モード/2x2 パターンディザに設定
ディザパターン値 A	output.dithering.dithering_pattern_a	IMGC_DITHER_2X2_PA	←	2x2 パターンディザのパターン値 A の設定 (dithering_mode が GLCDC_DITHERING_MODE_2X2PATTERN のときのみ有効)
ディザパターン値 B	output.dithering.dithering_pattern_b	IMGC_DITHER_2X2_PB	←	2x2 パターンディザのパターン値 B の設定 (dithering_mode が GLCDC_DITHERING_MODE_2X2PATTERN のときのみ有効)
ディザパターン値 C	output.dithering.dithering_pattern_c	IMGC_DITHER_2X2_PC	←	2x2 パターンディザのパターン値 C の設定 (dithering_mode が GLCDC_DITHERING_MODE_2X2PATTERN のときのみ有効)
ディザパターン値 D	output.dithering.dithering_pattern_d	IMGC_DITHER_2X2_PD	←	2x2 パターンディザのパターン値 D の設定 (dithering_mode が GLCDC_DITHERING_MODE_2X2PATTERN のときのみ有効)
輝度補正の有効、無効	output.brightness.enable	IMGC_BRIGHT_OUTCTL_ACTIVE	←	輝度補正を有効/無効に設定 (無効の場合 glcdc_cfg_t.output.brightness 以下の構造体メンバの設定値に依らず、RGB 信号の輝度調整値に 0 が設定されます) ※QE for Display [RX]では、常に true が設定されます。
R 信号の輝度調整値	output.brightness.r	IMGC_BRIGHT_OUTCTL_OFFSET_R	←	R 信号の輝度調整値の設定
G 信号の輝度調整値	output.brightness.g	IMGC_BRIGHT_OUTCTL_OFFSET_G	←	G 信号の輝度調整値の設定
B 信号の輝度調整値	output.brightness.b	IMGC_BRIGHT_OUTCTL_OFFSET_B	←	B 信号の輝度調整値の設定
コントラスト補正有効、無効	output.contrast.enable	IMGC_CONTRAST_OUTCTL_ACTIVE	←	コントラスト補正を有効/無効に設定 (無効の場合 glcdc_cfg_t.output.contrast 以下の構造体メンバの設定値に依らず、RGB 信号のコントラスト調整値に 1.000 が設定されます) ※QE for Display [RX]では、常に true が設定されます。

概略	構造体メンバ	QE for Display [RX]が出力する define 定義	サンプルの設定値	設定内容
R 信号のコントラスト調整値	output.contrast.r	IMGC_CONTRAST_OUT CTL_GAIN_R	←	R 信号のコントラスト調整値を設定
G 信号のコントラスト調整値	output.contrast.g	IMGC_CONTRAST_OUT CTL_GAIN_G	←	G 信号のコントラスト調整値を設定
B 信号のコントラスト調整値	output.contrast.b	IMGC_CONTRAST_OUT CTL_GAIN_B	←	B 信号のコントラスト調整値を設定
ガンマ補正の有効、無効	output.gamma.enable	IMGC_GAMMA_ACTIVE	←	ガンマ補正を有効/無効に設定 (無効の場合、 glcdc_cfg_t.output.gamma 以下の構造体メンバの設定値は無視されます) ※QE for Display [RX]では、常に true が設定されます。
R 信号のガンマ補正テーブル	output.gamma.p_r	・ gain[16] IMGC_GAMMA_R_GAIN_00~ IMGC_GAMMA_R_GAIN_15 ・ Threshold[15] IMGC_GAMMA_R_TH_01~ IMGC_GAMMA_R_TH_15	← (&gs_gamma_table_r)	R 信号の各領域のゲインと開始しきい値の設定  構造体メンバには、テーブル変数 (gs_gamma_table_r) のアドレスを設定しています。
G 信号のガンマ補正テーブル	output.gamma.p_g	・ gain[16] IMGC_GAMMA_G_GAIN_00~ IMGC_GAMMA_G_GAIN_15 ・ Threshold[15] IMGC_GAMMA_G_TH_01~ IMGC_GAMMA_G_TH_15	← (&gs_gamma_table_g)	G 信号の各領域のゲインと開始しきい値の設定  構造体メンバには、テーブル変数 (gs_gamma_table_g) のアドレスを設定しています。
B 信号のガンマ補正テーブル	output.gamma.p_b	・ gain[16] IMGC_GAMMA_B_GAIN_00~ IMGC_GAMMA_B_GAIN_15 ・ Threshold[15] IMGC_GAMMA_B_TH_01~ IMGC_GAMMA_B_TH_15	← (&gs_gamma_table_b)	B 信号の各領域のゲインと開始しきい値の設定  構造体メンバには、テーブル変数 (gs_gamma_table_b) のアドレスを設定しています。
CLUT メモリの更新の有効、無効の選択	clut.enable	-	グラフィック 2: false  グラフィック 1:未設定	CLUT メモリを更新する/しない (更新しない場合、clut 以下の構造体メンバの設定値は無視されます)



概略	構造体メンバ	QE for Display [RX]が出力する define 定義	サンプルの設定値	設定内容
CLUT メモリの先頭アドレスへのポインタ	clut.p_base	-	グラフィック 2: FIT_NO_PTR  グラフィック 1:未設定	ポインタが指し示すアドレスから値を読み出し CLUT メモリにコピーします
更新する CLUT メモリの開始エントリ番号	clut.start	-	グラフィック 2: 0  グラフィック 1:未設定	指定したエントリ番号から CLUT メモリの更新を開始します
更新する CLUT メモリのエントリサイズ	clut.size	-	グラフィック 2: 256  グラフィック 1:未設定	指定したサイズ分の CLUT メモリを更新します
VPOS 検出の許可、禁止	detection.vpos_detect	-	true	VPOS 検出を許可/禁止に設定
GR1UF 検出の許可、禁止	detection.gr1uf_detect	-	false	GR1UF 検出を許可/禁止に設定
GR2UF 検出の許可、禁止	detection.gr2uf_detect	-	true	GR2UF 検出を許可/禁止に設定
VPOS 割り込みの許可、禁止	interrupt.vpos_enable	-	true	VPOS 割り込みを許可/禁止に設定
GR1UF 割り込みの許可、禁止	interrupt.gr1uf_enable	-	false	GR1UF 割り込みを許可/禁止に設定
GR2UF 割り込みの許可、禁止	interrupt.gr2uf_enable	-	true	GR2UF 割り込みを許可/禁止に設定

— : 対応する定義なし

← : QE for Display [RX] が出力した定義を設定

## 7.4 周辺機能

本サンプルで使用する周辺機能を表 7-2 に示します。

表 7-2 使用する周辺機能

機能	用途
グラフィック LCD コントローラ (GLCDC)	ディスプレイ表示 <ul style="list-style-type: none"> <li>・グラフィック 2 を使用</li> <li>・グラフィック 1 は未使用</li> <li>・カラーフォーマット RGB565</li> <li>・出力データフォーマット RGB565 (パラレル 16 ビット)</li> </ul>
割り込みコントローラ (ICU)	GLCDC の割り込み制御
拡張 RAM	フレームバッファに使用

## 7.5 メモリ

本サンプルで使用する ROM/RAM サイズを表 7-3 に示します。

表 7-3 ROM/RAM 使用量

プロジェクト	使用メモリ	サイズ	備考
RSK RX72N	ROM	17266 バイト	コード、定数データ、初期値データ
	RAM	8129 バイト	データ : 376 バイト 未初期化データ : 2633 バイト STACK : 5120 バイト
Envision RX72N	ROM	17270 バイト	コード、定数データ、初期値データ
	RAM	8129 バイト	データ : 376 バイト 未初期化データ : 2633 バイト STACK : 5120 バイト
RSK RX65N	ROM	16842 バイト	コード、定数データ、初期値データ
	RAM	8013 バイト	データ : 376 バイト 未初期化データ : 2517 バイト STACK : 5120 バイト
Envision RX65N	ROM	16862 バイト	コード、定数データ、初期値データ
	RAM	8013 バイト	データ : 376 バイト 未初期化データ : 2517 バイト STACK : 5120 バイト

## 7.6 メモリマップ

本サンプルで使用する固有セクションを表 7-4 に示します。

表 7-4 サンプルで使用する固有セクション

セクション	タイプ	内容
FRAME_BUFFER	data	表示データが格納されるバッファ 拡張 RAM 領域の固定アドレス (0x00800000) に配置。

## 7.7 使用割り込み一覧

表 7-5 にサンプルで使用する割り込みを示します。これらの割り込みは GLCDC FIT モジュールで制御されています。詳細は GLCDC FIT モジュールのマニュアルを参照してください。

表 7-5 サンプルで使用する割り込み

割り込み	優先度	説明
VPOS 割り込み	5	グラフィック 2 ライン検出によって発生する割り込み (グループ AL1 割り込み 要因番号:8)
GR1UF 割り込み		バスへのアクセス負荷などにより、グラフィック 1 アルファブレンド部へグラフィックデータの供給が間に合わないときに発生する割り込み (グループ AL1 割り込み 要因番号:9) 本サンプルではグラフィック 1 を使用しないため割り込み禁止に設定。
GR2UF 割り込み		バスへのアクセス負荷などにより、グラフィック 2 アルファブレンド部へグラフィックデータの供給が間に合わないときに発生する割り込み (グループ AL1 割り込み 要因番号:10)

## 7.8 本サンプルの使用に際して

本サンプルは、GLCDC を初期化して画面を表示させるために必要な処理を main.c ファイル内に記述しています。ユーザシステムで参考にされる場合には、main.c ファイル内の下記変数や関数と同等のものがが必要です。また、main.c ファイル内で宣言している define 定義は、GLCDC FIT モジュールのパラメータへ代入する値として使用しているものもありますが、必須ではありません。define 定義の代わりに即値を代入しても問題ありません。

以下の内容を確認のうえ、ユーザシステムの参考にしてください。

### 7.8.1 変数一覧

表 7-6 に必要な変数を示します。

表 7-6 必要な変数

型	変数名	内容
static glcdc_cfg_t	gs_glcde_init_cfg	GLCDC FIT モジュールの R_GLCDC_Open 関数に渡す構造体変数です。 画面表示に必要な情報を設定します。
static volatile bool	gs_first_interrupt_flag	コールバック関数で初回の VPOS 割り込みを確認する変数です。
static const gamma_correction_t	gs_gamma_table_r = {gain[16]={IMGC_GAMMA_R_GAIN_00~ IMGC_GAMMA_R_GAIN_15} , Threshold[15]={IMGC_GAMMA_R_TH_01~ IMGC_GAMMA_R_TH_15}}	ガンマ（赤）テーブル テーブルデータは QE for Display [RX] から出力されるヘッダファイル (r_image_config_(ボード名).h) の define 定義を使用しています。
static const gamma_correction_t	gs_gamma_table_g = {gain[16]={IMGC_GAMMA_G_GAIN_00~ IMGC_GAMMA_G_GAIN_15} , Threshold[15]={IMGC_GAMMA_G_TH_01~ IMGC_GAMMA_G_TH_15}}	ガンマ（緑）テーブル テーブルデータは QE for Display [RX] から出力されるヘッダファイル (r_image_config_(ボード名).h) の define 定義を使用しています。
static const gamma_correction_t	gs_gamma_table_b = {gain[16]={IMGC_GAMMA_B_GAIN_00~ IMGC_GAMMA_B_GAIN_15} , Threshold[15]={IMGC_GAMMA_B_TH_01~ IMGC_GAMMA_B_TH_15}}	ガンマ（青）テーブル テーブルデータは QE for Display [RX] から出力されるヘッダファイル (r_image_config_(ボード名).h) の define 定義を使用しています。

## 7.8.2 関数一覧

表 7-7 に必要な関数を示します。

表 7-7 必要な関数

関数名	概要
glcdc_initialize	GLCDC の初期化と動作開始の関数です。
qe_for_display_parameter_set	QE for Display [RX]で設定する GLCDC FIT モジュールのパラメータをまとめた関数です。
glcdc_callback	GLCDC FIT モジュールのコールバック関数です。 GLCDC の割り込みを使用しない場合は必須ではありません。 使用しない場合は、GLCDC FIT モジュールのマニュアルに従い、割り込みの禁止設定と、コールバック関数へのポインタを無効 (FIT_NO_FUNC)に設定してください。
board_port_setting	ボード依存の端子設定 (LCD パネルのバックライト、リセット制御用端子)関数です。 使用する LCD パネルや接続に依存するため、必須ではありません。 RSK および Envision では、LCD パネルのバックライトとリセットを汎用入出力ポートで制御する接続になっています。

## 8. QE for Display [RX]の使用方法

本章では実際のディスプレイ調整の流れに従い QE for Display [RX]の使用方法について説明します。QE for Display [RX]の機能詳細につきましては、QE for Display [RX]に同梱のヘルプを参照ください。

### 8.1 QE for Display[RX]の起動

e<sup>2</sup> studio のメニューから『Renesas Views』→『Renesas QE』→『ディスプレイ調整 RX (QE)』を選択すると QE for Display[RX]が起動します(図 8-1)。

図 8-1 に示すブロック図は、GLCDC の H/W ブロック図を表しており、画像データが出力されるまでの経路と、各種画質補正が実施される位置関係を知ることができます。また、ブライトネス、コントラストなどの画質調整内容をクリックすることで、画質調整タブに遷移し、各種調整が可能です。

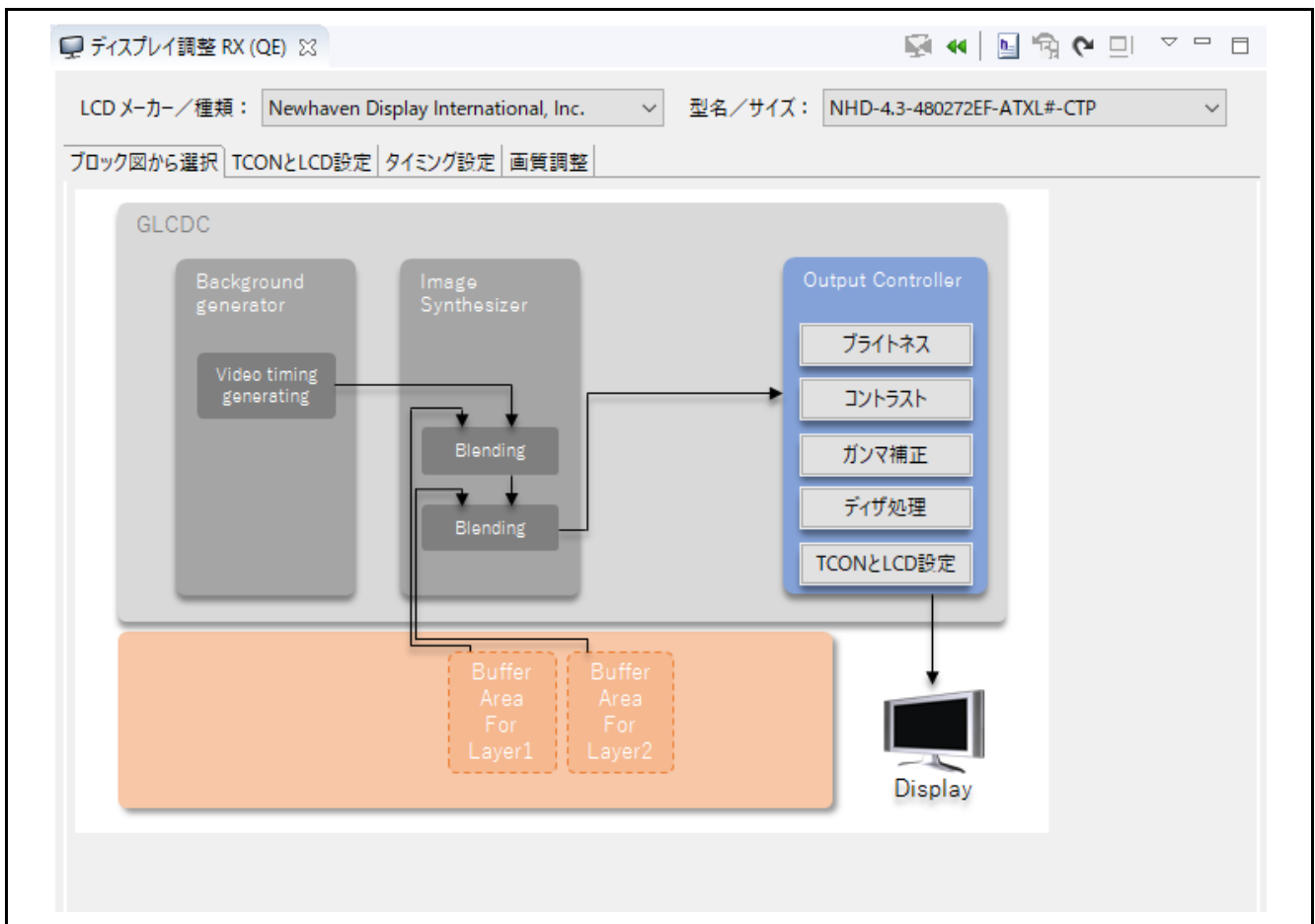


図 8-1 QE for Display[RX]の起動画面

## 8.2 LCD パネルデータの設定

ユーザが開発するシステムに接続する LCD パネルの情報を設定します。開発システムにディスプレイを接続するには LCD パネルの仕様とディスプレイコントローラの仕様を摺り合せ、設定可能かつ最適な設定値を見つけ出す必要があります。設定された情報はその際の比較情報として使用されます。

RSK に搭載の LCD は、Newhaven Display International 社製 NHD-4.3-480272EF-ATXL#-CTP です。また Envision に搭載の LCD は、EastRising 社製 ER-TFT043-3 です。

QE for Display[RX] V1.1.0 のパッケージには、該当の LCD パネルの情報が含まれていますのでそれを選択します。

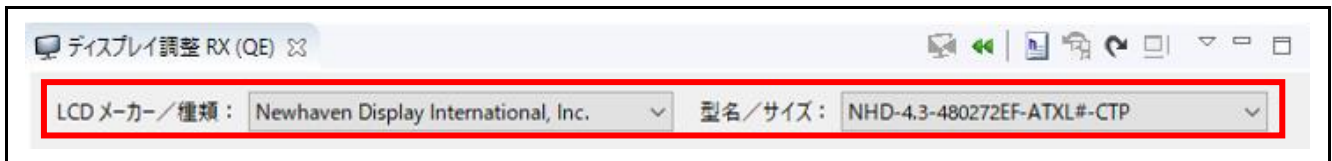


図 8-2 LCD パネル選択

LCD パネルの情報設定では表示方式の設定機能があります。表示方式は 3 つのパターンより選択し設定します。RSK および Envision 搭載の LCD パネルは表示方式 3(Vsync 信号、Hsync 信号、DE 信号を使用する方式)を採用しています。

LCD パネルの情報設定方法詳細については、9. LCD パネルデータの設定詳細 を参照してください。

## 8.3 制御信号の出力設定

QE for Display[RX]の[TCON と LCD 設定]タブを選択し、制御信号の設定を行います(図 8-3)。

このダイアログでは、以下に記載する制御信号の出力設定が可能です。

[パネルドライバ信号(TCON)の出力選択]

出力端子の選択 :

LCD\_TCON0~LCD\_TCON3 端子へ出力(TCON0~TCON3)

制御信号の極性 :

正極性(High Active)

負極性(Low Active)

[LCD の設定]

出力データフォーマットの選択 :

24bit RGB888 出力 (24bit(GLCDC\_OUT\_FORMAT\_24BITS\_RGB888))

18bit RGB666 出力 (18bit(GLCDC\_OUT\_FORMAT\_18BITS\_RGB666))

16bit RGB565 出力 (16bit(GLCDC\_OUT\_FORMAT\_16BITS\_RGB565))

データの出カタイミング :

パネルクロックの立ち上がりで出力 (Rising(GLCDC\_SIGNAL\_SYNC\_EDGE\_RISING))

パネルクロックの立ち下がりで出力 (Falling(GLCDC\_SIGNAL\_SYNC\_EDGE\_FALLING))

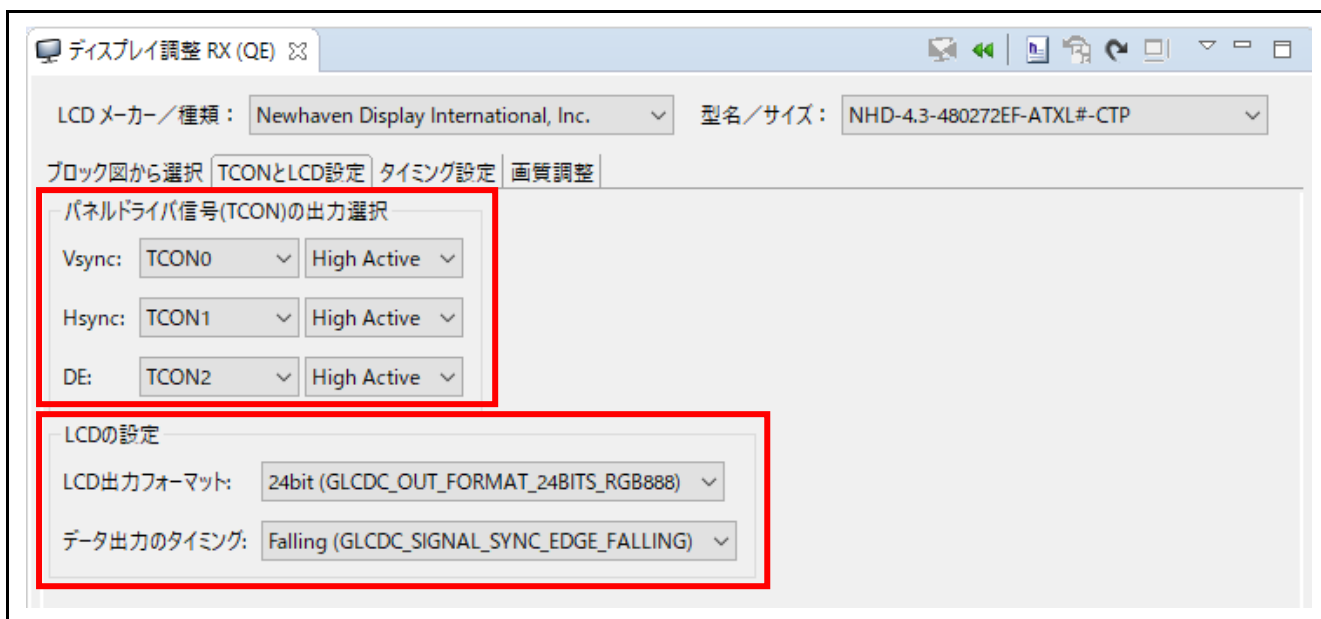


図 8-3 TCON と LCD 設定

各 RSK および Envision での設定はボード仕様より以下となります。

表 8-1 サンプルで使用する LCD パネル

	RSK RX72N / RX65N	Envision RX72N / RX65N
出力端子の選択		
Vsync	TCON0	TCON0
Hsync	TCON2	TCON2
DE	TCON3	TCON3
制御信号の極性		
Vsync	負極性(Low Active)	負極性(Low Active)
Hsync	負極性(Low Active)	負極性(Low Active)
DE	正極性(High Active)	正極性(High Active)
出力データフォーマットの選択	16bit RGB565 出力 (16bit(GLCDC_OUT_FORMAT_16BITS_RGB565))	16bit RGB565 出力 (16bit(GLCDC_OUT_FORMAT_16BITS_RGB565))
データの出力タイミング	パネルクロックの立ち上がりで出力 (Rising(GLCDC_SIGNAL_SYNC_EDGE_RISING))	パネルクロックの立ち上がりで出力 (Rising(GLCDC_SIGNAL_SYNC_EDGE_RISING))



## 8.4 LCD パネル制御信号タイミングの調整

デバッグを接続しサンプルプログラムを実行後、図 8-4 に示す設定値を変更することで制御信号タイミングの変更が可能です。変更した設定値を本ツールより GLCDC のレジスタに直接書き込み、RSK 上の LCD パネルに反映できます。

	値	規定値	差分
リフレッシュレート[Hz]	59.9	59.9	0.0
水平周波数[kHz]	17.3	17.1	0.2

(左: 設定値をレジスタに設定 右: 変更をリアルタイムでレジスタへ設定)

図 8-4 制御信号タイミングの調整

まずパネルクロック周波数をタイミング調整エリア左上のボックスに入力します。入力したパネルクロック周波数はリフレッシュレートの算出に使用され、画面下部に LCD パネル推奨値との差分として表示されます。パネルクロック周波数はプログラム中で設定する必要があります。QE for Display[RX]から設定することはできません。初期値は LCD パネルの推奨値が設定されています。ここではサンプルプログラム中で設定している 10MHz を入力します。

次に各パラメータを調整します。調整値が赤く表示された場合は、GLCDC の仕様または LCD パネル仕様の範囲外の値であることを意味します。この場合、GLCDC の仕様範囲内の値となるよう調整してください。GLCDC の仕様が許容する値の範囲は、赤く表示された調整値へマウスをポイントすることで確認することができます。

これらの初期値も LCD パネルの推奨値が使用されます。LCD パネルの水平フロントポーチ(HFP)の推奨値は 2 となっており、GLCDC の仕様範囲から外れているため変更が必要です。よって LCD パネル、GLCDC の仕様を満たす 3 以上へ変更します。これにより、調整値の表示が赤から黒に変わります。



調整値を決定したところで、本ツール上から GLCDC のレジスタに調整値を書き込み、結果を確認します。



図 8-5 レジスタ設定機能ボタン

レジスタへの反映は、下記の 2 つの方法があります。

表 8-2 レジスタへの反映機能

ボタン	名前	機能
	レジスタへ設定	設定値をレジスタに書き込みます。デバッグ接続時のみ有効です。
	変更をリアルタイムでレジスタへ設定	オンになっている時、設定値を変更する度にレジスタへ自動で書き込みます。デフォルトはオフ。 なお、デバッグに接続している場合のみ書き込み処理を行います。(デバッグ未接続時は何もしません)

注意：

QE for Display[RX]による調整値のレジスタ書き込み機能は、タイミング調整を行うと共にグラフィック画面をバックグラウンド画面の左上に調整する仕様となっています。

グラフィック画面およびバックグラウンド画面の定義につきましては、『RX65N グループ、RX651 グループ ユーザーズマニュアル ハードウェア編』(R01UH0590)または『RX72N グループ ユーザーズマニュアル ハードウェア編』(R01UH0824)を参照ください。

本サンプルプログラムでは、青いイメージの外周に赤い 1pixel 幅のラインの描画をします。サンプルプログラムでは RSK および Envision に合わせて最適な調整値が設定済みのため調整が不要ですが、実際の開発では赤い外周のラインが LCD パネルに表示されるように位置合わせを行います。

また、LCD パネルの表示方式および仕様により、各種設定値の細かな変更(数 pixel 単位での移動など)が LCD の表示として現れないことがあります。RSK および Envision 搭載の LCD パネルでは表示方式 3 を採用しており、バックポーチなどの設定値による表示位置の移動はできません。

## 8.5 制御信号出力／タイミング調整結果の反映

タイミング調整結果は、ヘッダファイルを出力することでプログラムに反映することができます。QE for Display[RX]のヘッダファイルの生成ボタン(図 8-6)を押下し、設定した制御タイミングを反映したヘッダファイルを生成します。



図 8-6 ヘッダファイルの生成ボタン

「タイミングおよび TCON 設定用ヘッダファイル」のみにチェックをつけ、「生成する」を選択すると、指定した出力先にヘッダファイルが生成されます。ヘッダファイル名と出力先は、任意で設定可能です。

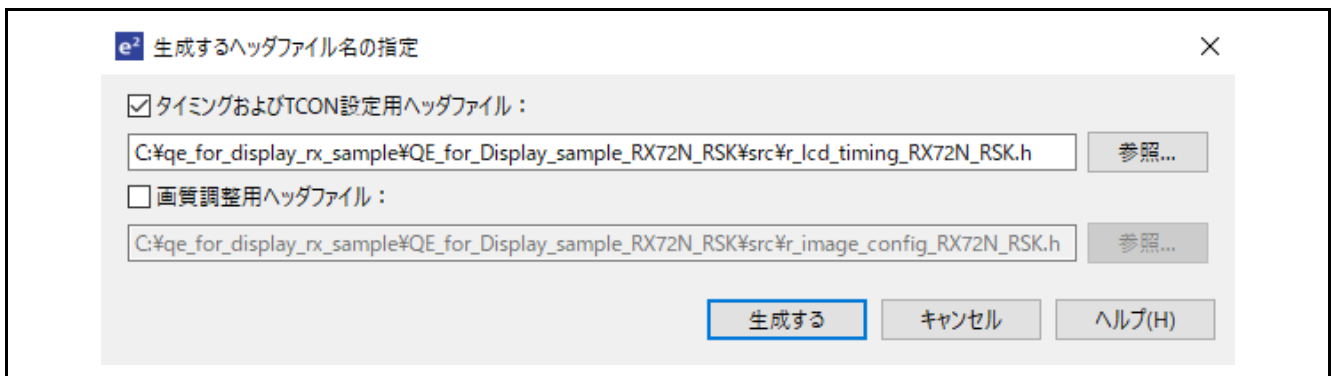


図 8-7 タイミングおよび TCON 設定用ヘッダファイルの生成

本サンプルプロジェクトに反映するには、下記フォルダに `r_lcd_timing_<RX72N/RX65N>_<RSK/Envision>.h` の名前で出力し、その後プロジェクトをクリーン、ビルドします。

フォルダ :

`<workspace folder> \%QE_for_Display_sample_<RX72N/RX65N>_<RSK/Envision>\src`

## 8.6 イメージダウンロード機能

QE for Display[RX]では LCD の特性に合わせて LCD の表示を確認しながら画質を調整します。その際にプログラムを変更せずに LCD に表示するイメージ変更することが可能です。

イメージダウンロード機能では、パソコン上にある LCD に表示するイメージデータ(バイナリファイル)をダウンロードすることができます。



図 8-8 イメージダウンロード機能ボタン

ツールバーのイメージダウンロード機能ボタンを押します。

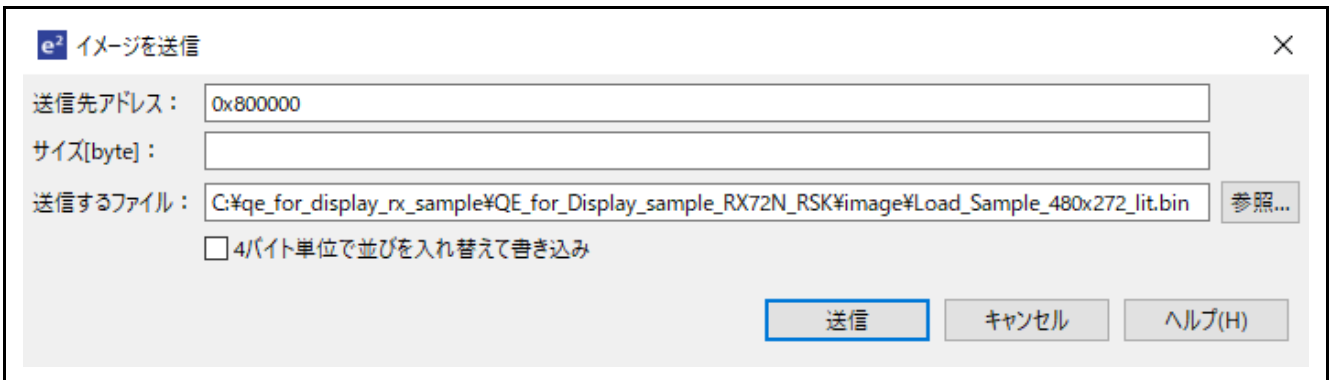


図 8-9 イメージを送信ダイアログ

表示された[イメージを送信]ダイアログに送信先アドレス、送信するファイルを指定します。[送信先アドレス]のデフォルトはグラフィック 2 フレームバッファ制御レジスタに設定されている値が設定されます。本サンプルプログラムでは、グラフィック 2 を使用していますので変更は不要です。[サイズ]の指定を省略した場合、[送信先アドレス]エディットボックスに指定されたアドレスから[送信するファイル] エディットボックスに指定されたファイル全体を書き込みます。

本アプリケーションノートには、サンプルのイメージデータが同梱されています。下記を送信します。

フォルダ :

`<workspace folder> %QE_for_Display_sample_< RX72N / RX65N >_< RSK / Envision >%image`

ファイル :

`Load_Sample_480x272_lit.bin`

正しく送信が完了すると、図 8-10 の様なカラーバーが表示されます。

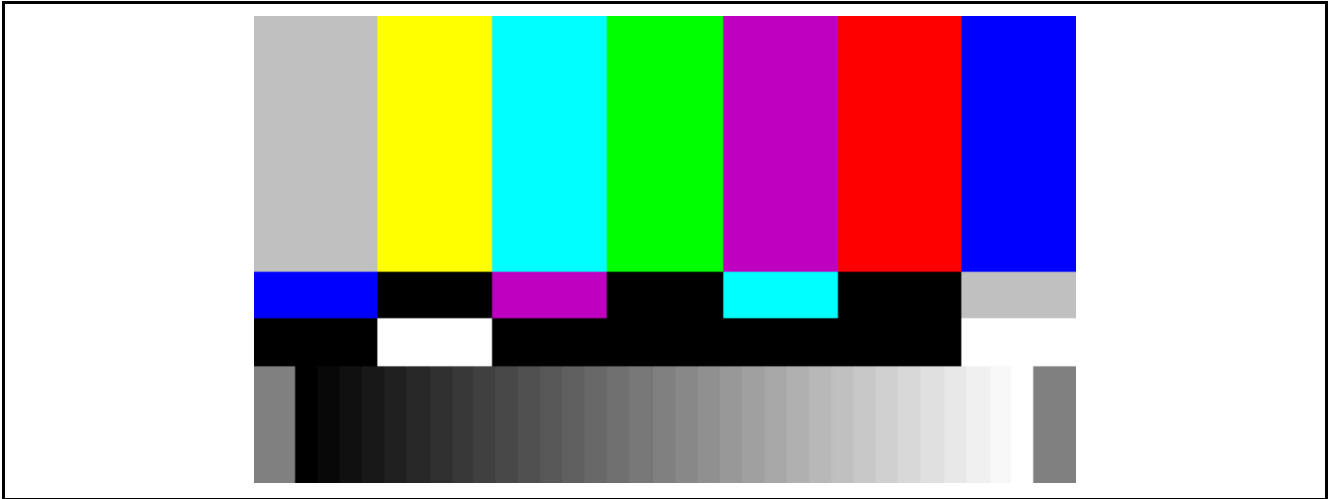


図 8-10 送信イメージ

## 8.7 各種画質の調整

[ブロック図から選択]タブの図 8-11 に赤枠で示す画質調整内容をクリックすることで、画質調整タブに遷移し、各種画質調整が可能です。

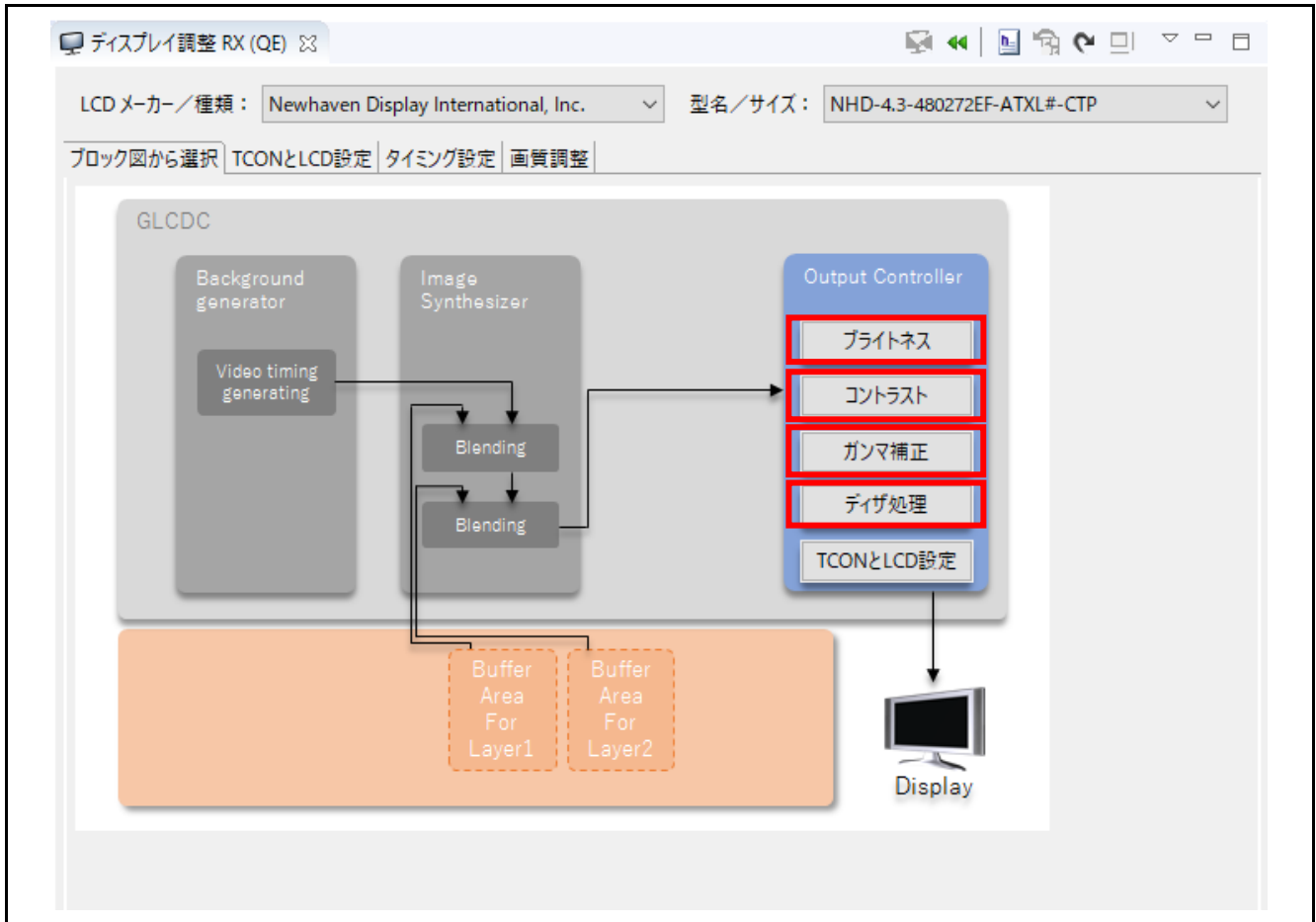


図 8-11 各種画質調整ボタンの選択

画質調整タブでは各種画質調整が可能です。QE for Display[RX]では「調整処理順設定」および「ブライトネス」、「コントラスト」、「ガンマ補正」、「ディザ処理」の4つの画質調整機能をサポートします。

これらの調整機能では設定の変更をリアルタイムで反映し、LCD パネルの表示を見ながら調整することができます。

画質調整は簡易設定、またはカスタムにて行います。カスタムで画質調整を行う場合は、『RX65N グループ、RX651 グループ ユーザーズマニュアル ハードウェア編』(R01UH0590)または『RX72N グループ ユーザーズマニュアル ハードウェア編』(R01UH0824)、および『RX ファミリ グラフィック LCD コントローラモジュール Firmware Integration Technology』(R01AN3609)を参照し、各レジスタで設定する content と、設定可能な値を確認の上、調整を行ってください。



図 8-12 画質調整画面

## 8.8 画質調整用ヘッダファイルの生成

QE for Display[RX]のヘッダファイルの生成アイコンを押下することで設定した画質調整内容を反映したヘッダファイルを生成する事が可能です。(図 8-13 参照)

「画質調整用ヘッダファイル」のみにチェックをつけ、「生成する」を選択すると、指定した出力先にヘッダファイルが生成されます。ヘッダファイル名と出力先は、任意で設定可能です。

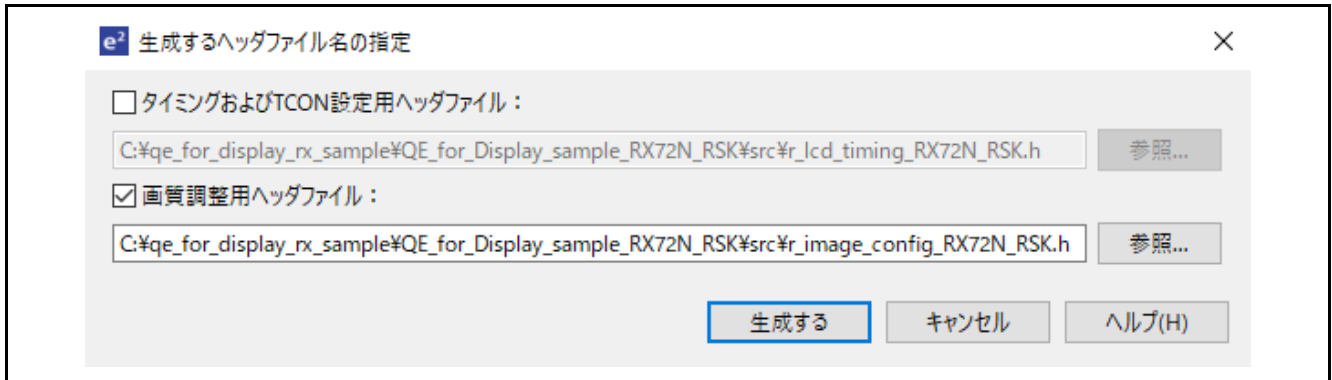


図 8-13 画質調整用ヘッダファイルの生成

本サンプルプロジェクトに反映するには、下記フォルダに `r_image_config_<RX72N/RX65N>_<RSK/Envision>.h` の名前で出力し、その後プロジェクトをクリーン、ビルドします。

フォルダ :

`<workspace folder> ¥QE_for_Display_sample_<RX72N/RX65N>_<RSK/Envision>¥src`



## 9. LCD パネルデータの設定詳細

図 8-1 のダイアログ上部にある[LCD メーカー/種類]のプルダウンリストから[カスタム]を選択するとカスタムディスプレイ・データの編集ダイアログ(図 9-1)が表示されます。このダイアログに LCD パネルの情報を入力します。

図 9-1 カスタムディスプレイ・データの編集ダイアログ

Parameter	Min.	Typ.	Max.
Panel Clock Frequency (PCF) [MHz]	-		-
Panel Clock Period (PCP) [ns]	-	-	-
Horizontal Frequency (HF) [KHz]	-	-	-
Horizontal Period (HP) [us]	-	-	-
Horizontal Total Period (HTP) [Clock]	-	-	-
Horizontal Pulse Width (HPW) [Clock]	-		-
Horizontal Display Period (HDP) [Clock]	-		-
Horizontal Front Porch (HFP) [Clock]	-		-
Horizontal Back Porch (HBP) [Clock]	-		-
Vertical Frequency (VF) [Hz]	-	-	-
Vertical Period (VP) [ms]	-	-	-
Vertical Total Period (VTP) [Line]	-	-	-
Vertical Pulse Width (VPW) [Line]	-		-
Vertical Display Period (VDP) [Line]	-		-
Vertical Front Porch (VFP) [Line]	-		-
Vertical Back Porch (VBP) [Line]	-		-

図 9-1 カスタムディスプレイ・データの編集ダイアログ

### 9.1 登録名称の記入

カスタムディスプレイ・データの編集ダイアログ(図 9-2)の『メーカー/種類』、『型名/サイズ』項目に、任意の名称を入力してください。この名称がドロップダウンリストに登録され、以後選択する事が可能になります。

図 9-2 名称の登録

図 9-2 名称の登録

## 9.2 表示方式の選択

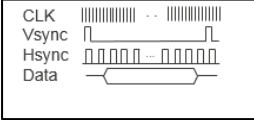
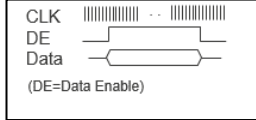
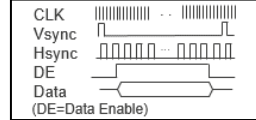
LCD パネルを接続する為に必要な主な制御信号を表 9-1 主な制御信号 に示します。QE for Display[RX] では、これらの制御信号を組み合わせた 3 つの方式の表示機器をサポートしています。

表 9-1 主な制御信号

名称	機能概略
水平同期信号(Hsync)	表示する 1 ラインのタイミングを作る信号
垂直同期信号(Vsync)	表示する 1 画面のタイミングを作る信号
パネルクロック(CLK)	表示される画素のサンプリングに使用される信号
表示イネーブル(DE)	有効なデータが出力されていることを示す信号
データ(Data)	表示するデータ

ユーザは使用する LCD パネルの仕様から、どの制御信号が必要であるかを確認し、表 9-2 使用する制御信号 に示す 3 つの表示方式から選択する必要があります。

表 9-2 使用する制御信号

名称	表示方式 1	表示方式 2	表示方式 3
			
水平同期信号(Hsync)	使用	未使用	使用
垂直同期信号(Vsync)	使用	未使用	使用
パネルクロック(CLK)	使用	使用	使用
表示イネーブル(DE)	未使用	使用	使用
データ(Data)	使用	使用	使用

### 9.3 制御タイミングの入力

LCD パネルのデータシートより制御タイミングを入力します。Typ.に入力された値がタイミング制御の初期値に使用され、Min、Max については、QE for Display[RX]のタイミング調整を GUI で行った時、範囲内であるかどうかの確認に使用されます。

図 9-3 は RSK 搭載の LCD パネルの場合のデータ入力結果です。表 9-3 LCD パネルのデータシートを参照し入力します。

The screenshot shows a GUI window titled "カスタムディスプレイデータの編集" (Custom Display Data Edit). It contains the following elements:

- メーカー/種類: メーカー1
- 型名/サイズ: LCDパネル1
- 表示方式: 表示方式3 (selected)
- Timing diagram showing CLK, Vsync, Hsync, DE, and Data signals.
- 既存データを流用 (Use existing data) button.
- Parameter table with columns: Parameter, Min., Typ., Max.
- URL input field.
- OK, キャンセル, ヘルプ(H) buttons.

Parameter	Min.	Typ.	Max.
Panel Clock Frequency (PCF) [MHz]	-	9.0	15.0
Panel Clock Period (PCP) [ns]	66.6666666666...	111.1111111111...	-
Horizontal Frequency (HF) [KHz]	-	17.14	-
Horizontal Period (HP) [us]	-	58.3430571761...	-
Horizontal Total Period (HTP) [Clock]	525	525	605
Horizontal Pulse Width (HPW) [Clock]	2	41	41
Horizontal Display Period (HDP) [Clock]	480	480	480
Horizontal Front Porch (HFP) [Clock]	2	2	82
Horizontal Back Porch (HBP) [Clock]	2	2	41
Vertical Frequency (VF) [Hz]	-	59.94	-
Vertical Period (VP) [ms]	-	16.6833500166...	-
Vertical Total Period (VTP) [Line]	285	286	399
Vertical Pulse Width (VPW) [Line]	1	10	11
Vertical Display Period (VDP) [Line]	272	272	272
Vertical Front Porch (VFP) [Line]	1	2	227
Vertical Back Porch (VBP) [Line]	1	2	11

空欄(黄色セル)は入力推奨欄です

図 9-3 制御タイミング入力結果

表 9-3 RSK 搭載 LCD パネルのデータシート抜粋

Parameter	Symbol	Spec.			Unit
		Min.	Typ.	Max.	
Clock cycle	fclk	-	9	15	MHz
Hsync cycle	1/th	-	17.14	-	KHz
Vsync cycle	1/th	-	59.94	-	Hz
Horizontal Signal					
Horizontal cycle	th	525	525	605	CLK
Horizontal display period	thd	480	480	480	CLK
Horizontal front porch	thf	2	2	82	CLK
Horizontal pulse width	thp	2	41	41	CLK
Horizontal back porch	thb	2	2	41	CLK
Vertical Signal					
Vertical cycle	tv	285	286	399	H
Vertical display period	tvd	272	272	272	H
Vertical front porch	tvf	1	2	227	H
Vertical pulse width	tvp	1	10	11	H
Vertical back porch	rvb	1	2	11	H

#### 9.4 作成したディスプレイデータの編集

ツールバーのメニューボタンをクリックし、[カスタムディスプレイの編集と削除...]メニューを実行することで、作成したディスプレイデータの再編集が行えます。

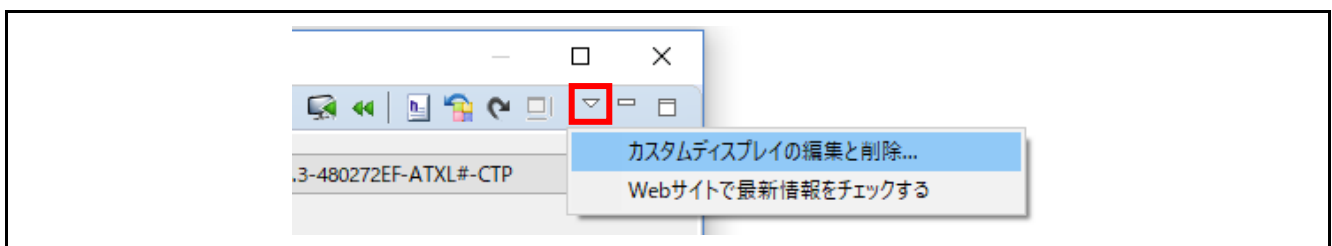


図 9-4 [カスタムディスプレイの編集と削除...]メニュー

## 10. ユーザ環境に適用するには

本サンプルをユーザ環境に適用するには、1.1 QE for Display [RX]を使った開発のフローの手順でサンプルプログラムを変更する必要があります。本章では、変更点および各手順における補足説明および注意事項を示します。

以降、RSK RX72N を例として説明します。

ユーザ環境に応じて変更するポイント

### <プロジェクトの設定>

- Smart Configurator の選択（新規プロジェクト作成時）
- セクションの設定

### <SmartConfigurator の設定>

- クロックの設定
- GLCDC FIT モジュールの追加
- GLCDC で使用する端子の設定

### <プログラムの作成（main.c ファイルの変更）>

- パネルクロック
- 端子割り付け（ビットエンディアン）
- ピクセル配列
- その他の制御端子（バックライト、リセットなど）
- ヘッダファイルの名称
- グラフィック 2 の表示サイズ
- グラフィック 2 のベースアドレス

## 10.1 仕様確認

GLCDC に設定するパラメータの内、QE for Display [RX] でサポートしていないパラメータはユーザが設定する必要があります。（7.3 GLCDC FIT モジュールのパラメータと QE for Display [RX] が出力するヘッダとの対応を参照。）また、ユーザボードに応じて別途 LCD パネルの制御が必要です。

以下の設定はユーザ環境のボード仕様をもとに決定してください。

- パネルクロック
- 端子割り付け（ビットエンディアン）
- ピクセル配列
- その他の制御端子（バックライト、リセットなど）

### 10.1.1 パネルクロック

GLCDC では、クロックソース（PLL）を分周（1 分周～32 分周）した周波数をパネルクロックとして使用します。

RSK RX72N に搭載されている LCD パネルの入カクロックは、9MHz（Typ.）～ 15MHz（Max）です。本サンプルでは、クロックソースに対して 24 分周しています。

$$\text{PLL}[240\text{MHz}](*) \div 24 \text{ 分周} = 10\text{MHz} \quad (※) \text{ PLL} = \text{EXTAL}[24\text{MHz}] \times 10 \text{ 通倍} \times 1 = 240\text{MHz}$$

```
/* Output clock */
gs_glcddc_init_cfg.output.clock_div_ratio = GLCDC_PANEL_CLK_DIVISOR_24;
```

図 10-1 パネルクロックの設定（クロックソースからの分周比）

結果、本サンプルではパネルクロックを 10MHz で出力しています。

LCD パネルの Typ. が 9MHz に対して 1MHz の差分がありますので、QE for Display [RX] を使用して水平周波数やリフレッシュレートの調整が必要になってきます。

### 10.1.2 端子割り付け(ビットエンディアン)とピクセル配列

GLCDC では、マイコンと LCD パネルとの接続に合わせて、端子割り付け（「リトルエンディアン」または「ビッグエンディアン」）とピクセル配列（「RGB」または「BGR」）をそれぞれ選択します。

RSK RX72N に搭載されている LCD パネルとマイコンとの接続は、6.2 端子機能に記載のとおりで、端子割り付け（ビットエンディアン）は「リトルエンディアン」になります。また、ピクセル配列は「RGB」となります。

上記は出力データフォーマットごとに異なります。出力データフォーマットごとの詳細は、対象マイコンのユーザーズマニュアル ハードウェア編を参照ください。

```
/* Endian */
gs_glcddc_init_cfg.output.endian = GLCDC_ENDIAN_LITTLE;

/* Color order */
gs_glcddc_init_cfg.output.color_order = GLCDC_COLOR_ORDER_RGB;
```

図 10-2 端子割り付け(ビットエンディアン)とピクセル配列の設定

### 10.1.3 その他の制御端子

ユーザボードにおける LCD パネルとマイコンの接続に応じて、バックライトやリセットの制御が必要な場合があります。

RSK RX72N では、6.2 端子機能に記載のとおり、LCD パネルのバックライトやリセットをマイコンの汎用入出力ポートで制御しています。

```
static void board_port_setting (void)
{
    /* ---- Port setting ---- */
    /* LCD back light and display-on */
    PORT2.PODR.BIT.B7 = 1; /* Back light */
    PORTK.PODR.BIT.B4 = 1; /* Display */
    PORT2.PDR.BIT.B7 = 1;
    PORTK.PDR.BIT.B4 = 1;

} /* End of function board_port_setting() */
```

図 10-3 バックライトとリセットの制御

## 10.2 プロジェクトの作成

### 10.2.1 Smart Configurator の選択

ユーザが使用するマイコンで新規プロジェクトを作成します。e<sup>2</sup> studio の画面に従って進めると、図 10-4 の画面が表示されますので、[スマート・コンフィグレータを使用する] にチェックを入れてください。



図 10-4 コーディング・アシストツールの選択



### 10.2.2 セクションの設定

フレームバッファとして使用する領域にセクションを設定します。プロジェクトのプロパティを開いて、FRAME\_BUFFER セクションを追加してください。サンプルでは、この FRAME\_BUFFER セクションが配置されている領域に初期画面を生成しています。また、グラフィック 2 のベースアドレスには、このセクションの先頭アドレスを指定しています。

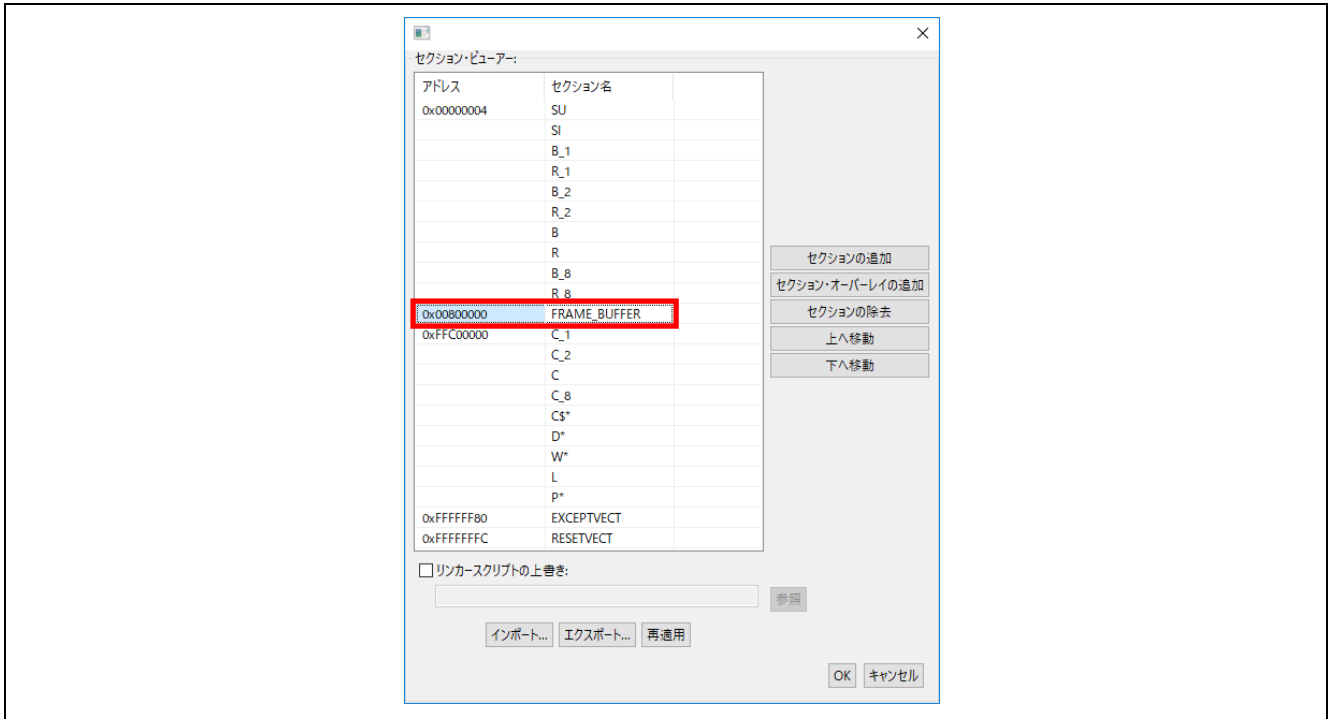


図 10-5 セクションの設定

なお、本サンプルで使用する拡張 RAM 領域（7.6 メモリマップ）は RX72N グループの場合 512K バイトのため、ユーザ環境で使用する LCD パネルの表示サイズによっては内蔵 ROM への配置を検討する必要があります。その際は、プログラムと合わせて画像データも内蔵 ROM にダウンロードしなければならないため、サンプルの初期画面は使用できません。また、QE for Display [RX] のイメージダウンロード機能も内蔵 ROM へのダウンロードはできません。内蔵 ROM の容量については、対象マイコンのユーザーズマニュアル ハードウェア編を確認してください。

### 10.3 Smart Configurator の設定

Smart Configurator を使用してクロックの設定やコンポーネント、端子の設定を行います。全ての設定が完了したら、コードの生成ボタンをクリックしてコードの生成を行ってください。Smart Configurator の詳細は Smart Configurator のマニュアルを参照してください。

#### 10.3.1 クロックの設定

Smart Configurator の[クロック]タブより、ユーザ環境に合わせてクロックの設定を行ってください。また、PLL はパネルクロックのクロックソースになりますので周波数を間違えないようご注意ください。

#### 10.3.2 GLCDC FIT モジュールの追加

Smart Configurator の[コンポーネント]タブより、GLCDC FIT モジュールをプロジェクトに追加してください。コンポーネントの追加ボタンをクリックし、[コンポーネントの追加]ダイアログのコンポーネント一覧から GLCDC FIT モジュール (r\_glcdc\_rx) を選択します。モジュールのダウンロードがまだの場合は、このダイアログの[他のソフトウェアコンポーネントをダウンロードする]からダウンロードできます。

#### 10.3.3 GLCDC で使用する端子の設定

Smart Configurator の[コンポーネント]タブより、追加された GLCD FIT モジュール (r\_glcdc\_rx) を選択すると、GLCDC で使用可能な端子の一覧が表示されます。ユーザ環境に合わせて各端子を設定してください。

[コンポーネント]タブの入力後、[端子]タブで各端子機能に割り当てるポート番号を指定してください。

RSK RX72N のサンプルでは、7.2.3 GLCDC の端子設定のとおり設定しています。

### 10.4 QE for Display [RX]による調整(初期設定)

8.QE for Display [RX]の使用方法与 9. LCD パネルデータの設定詳細の内容をもとに、QE for Display [RX]を使って、ユーザ環境で使用する LCD パネルの情報を入力し、最初の調整を行います。

調整の際は、GLCDC FIT モジュールの設定範囲に注意してください。QE for Display [RX]の[タイミング設定]タブにおける各設定値は、登録したパネルデータおよび GLCDC の設定範囲に違反すると赤く表示されます。そのため、黒く表示されている状態であっても、GLCDC FIT モジュールの設定範囲に違反している場合があります。

調整が完了したらヘッダファイルを生成します。

RSK RX72N のサンプルでは、QE\_for\_Display\_sample\_RX72N\_RSK¥src フォルダ直下に

r\_lcd\_timing\_RX72N\_RSK.h

r\_image\_config\_RX72N\_RSK.h

の名称でヘッダファイルを生成しています。

## 10.5 プログラムの作成

本サンプルをベースにプログラムを作成します。

### 10.5.1 サンプルプログラムのコピー

プロジェクト生成時に作成された「新規プロジェクト名」.c ファイルを削除し、本サンプルの main.c ファイルを新規作成したプロジェクトにコピーします。

### 10.5.2 プログラムの変更

7.8 本サンプルの使用に際して や 10.1 仕様確認で確認した仕様を参考に main.c ファイルを変更し、プログラムを完成させます。このとき、下記の内容について変更が必要な場合があります。

#### (a) ヘッダファイルの名称

本サンプルの main.c ファイルでインクルードしているヘッダファイルを QE for Display [RX] で生成したヘッダファイルに変更してください。

```
/* Header files for RSKRX72N board output by QE for Display [RX] */
#include "r_image_config_RX72N_RSK.h"
#include "r_lcd_timing_RX72N_RSK.h"
```

図 10-6 ヘッダファイルのインクルード

#### (b) グラフィック 2 の表示サイズ

使用する LCD パネルの表示サイズに合わせて define 定義の値を変更してください。この define 定義を本サンプルで表示される初期画面のデータ生成にも使用しているため、定義値の変更に合わせて初期画面の表示サイズも変更されます。(main.c の frame\_buffer\_initialize 関数)

```
/* Image info definition */
#define IMAGE_WIDTH      (480u) /* Width of image used in this sample. */
#define IMAGE_HEIGHT     (272u) /* Height of image used in this sample. */

/* ---- Graphic 2 setting ---- */
/* Image format */
gs_glcde_init_cfg.input[GLCDC_FRAME_LAYER_2].hsize      = IMAGE_WIDTH;
gs_glcde_init_cfg.input[GLCDC_FRAME_LAYER_2].vsize      = IMAGE_HEIGHT;
```

図 10-7 グラフィック 2 の画像サイズ指定

#### (c) グラフィック 2 のベースアドレス

FRAME\_BUFFER セクションの領域を使用しない、ROM に配置したデータを表示したいなどの場合は、グラフィック 2 のベースアドレスを、画像データが格納されている領域の先頭アドレスに変更してください。

```
/* ---- Graphic 2 setting ---- */
/* Image format */
gs_glcde_init_cfg.input[GLCDC_FRAME_LAYER_2].p_base = (uint32_t *) FRAME_BUF_BASE_ADDR;
```

図 10-8 グラフィック 2 のベースアドレス指定

## 10.6 実行から調整終了まで

プログラムの作成が完了したらデバッグを起動してプログラムを実行してください。初期画面が正常に表示されない場合、正しく設定されていません。QE for Display [RX] による調整値、および GLCDC FIT モジュールのパラメータ設定などを確認してください。

図 10-9 にトラブルシューティングを示します。

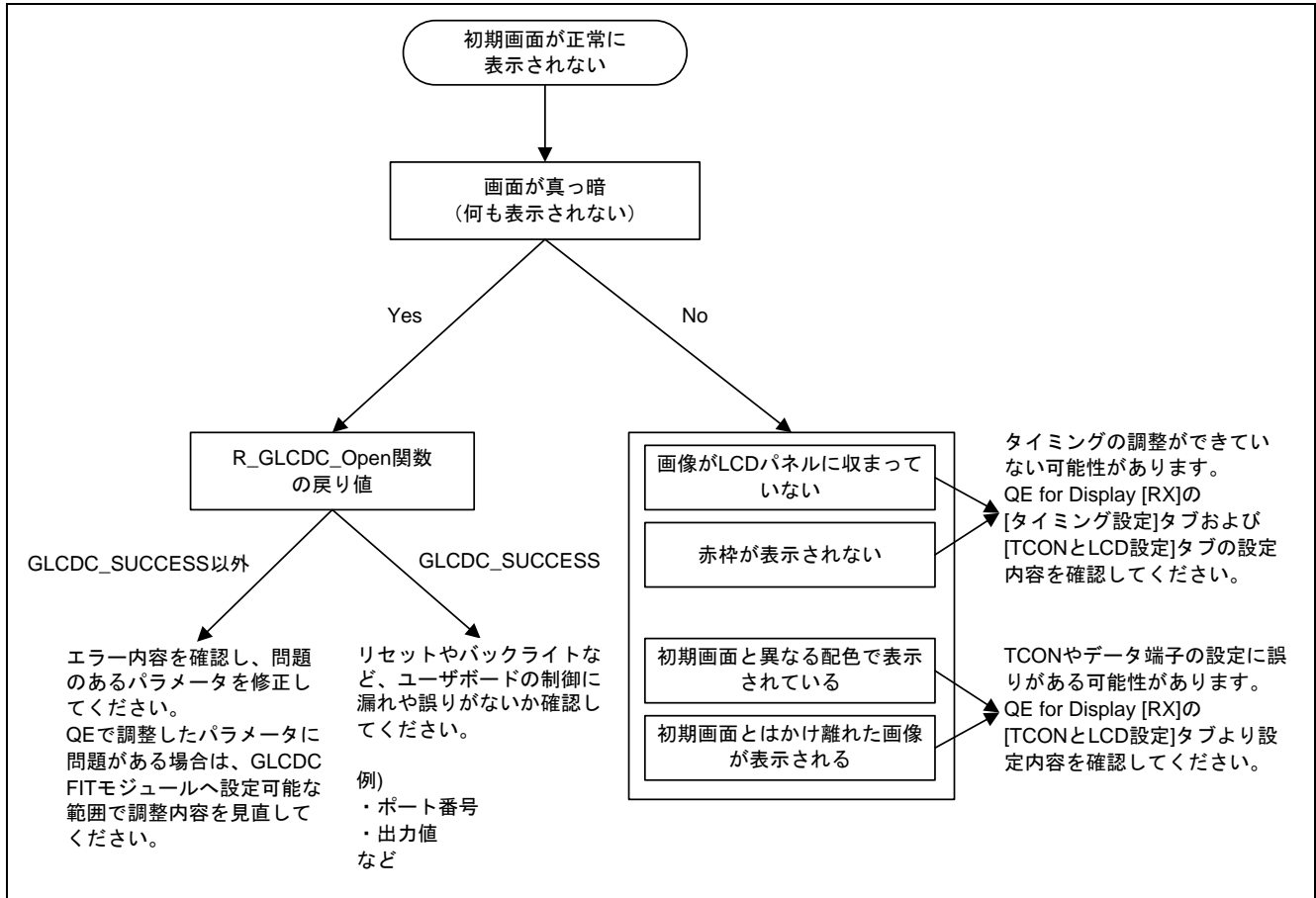


図 10-9 トラブルシューティング

## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	Jan.31.18	-	初版発行
1.10	Apr.07.20	-	RSK RX72N, Envision Kit RX72N のサポート追加
		11	「図 4-1 プロジェクトのフォルダ構成」変更
		14-19 38-52	図を e <sup>2</sup> studio v7.7.0 と RSK RX72N 用プロジェクトを使用した画面に変更
		26	表 7-1 「設定内容」欄にある、一部説明を変更
		34	「表 7-3 ROM/RAM 使用量」を更新

## 製品ご使用上の注意事項

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

### 1. 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っていると、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。

## 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）、交通管制（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、当社は一切その責任を負いません。

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

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

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

(Rev.4.0-1 2017.11)

## 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](http://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/](http://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。