

RX ファミリ

QE for Display モジュール Firmware Integration Technology

要旨

本アプリケーションノートでは、Firmware Integration Technology (以下、FIT)を使用した QE for Display モジュールについて説明します。本モジュールはスタンドアロン版 QE for Display[RX](以下、QE for Display)による GLCDC の調整に使用します。

e² studio およびプラグイン版 QE for Display[RX]を使用する場合、本モジュールは必要ありません。

また、本モジュールは GLCDC の調整専用のモジュールのため、調整後は本モジュールをアンインストールしていただくか、調整専用のプロジェクトを用意してご使用いただくことを推奨いたします。

QE for Display[RX,RA]の詳細は弊社ホームページを参照してください。

- ・ [ディスプレイ対応開発支援ツール QE for Display | Renesas](#)

対象環境

スタンドアロン版 QE for Display[RX]を使用する開発環境

(Renesas 製 CS+、IAR 社製 EWRX を想定)

対象デバイス

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

本アプリケーションノートを他のマイコンへ適用する場合、そのマイコンの仕様に併せて変更し、十分評価してください。

対象コンパイラ

- ・ Renesas Electronics C/C++ Compiler Package for RX Family
- ・ GCC for Renesas RX
- ・ IAR C/C++ Compiler for Renesas RX

各コンパイラの動作確認内容については 4.3 動作確認環境を参照してください。

関連ドキュメント

- Firmware Integration Technology ユーザーズマニュアル(R01AN1833)
- ボードサポートパッケージモジュール Firmware Integration Technology (R01AN1685)

目次

1. 概要	3
1.1 QE for Display FIT モジュールとは	3
1.2 QE for Display FIT モジュールの API 関数	3
1.3 制限事項	3
1.4 ソフトウェア構成	4
1.5 基本動作	4
2. API 情報	6
2.1 ハードウェアの要求	6
2.2 ソフトウェアの要求	6
2.3 サポートされているツールチェーン	6
2.4 ヘッダファイル	6
2.5 整数型	6
2.6 コンパイル時の設定	7
2.7 コードサイズ	8
2.8 戻り値	9
2.9 FIT モジュールの追加方法	9
2.10 for 文、while 文、do while 文について	10
3. API 関数	11
3.1 R_QE_DISPLAY_Open ()	11
3.2 R_QE_DISPLAY_Exec ()	12
3.3 R_QE_DISPLAY_GetVersion ()	13
4. 付録	14
4.1 サンプルコード	14
4.1.1 GLCDC FIT モジュールのみ使用する場合のサンプルコード	15
4.1.2 emWin FIT モジュールを使用する場合のサンプルコード	17
4.1.3 Aeropoint GUI FIT モジュールを使用する場合のサンプルコード	19
4.2 開発環境 (CS+、EWRX) の設定	21
4.2.1 マクロ定義の設定	21
4.2.2 デバッグ・コンソールの表示方法	22
4.3 動作確認環境	23
4.4 トラブルシューティング	24
5. 参考ドキュメント	25
改訂記録	26

1. 概要

1.1 QE for Display FIT モジュールとは

本モジュールを使用することでスタンドアロン版 QE for Display[RX]と通信して GLCDC のタイミング調整や画質の補正を行うことができます。

1.2 QE for Display FIT モジュールの API 関数

表 1.1 に本モジュールに含まれる API 関数を示します。

表 1.1 API 関数一覧

関数	関数説明
R_QE_DISPLAY_Open	GLCDC の初期化、および QE Display とのシリアル通信を初期化します。
R_QE_DISPLAY_Exec	QE Display からの受信データをもとに GLCDC の調整処理やエラー処理を実行します。
R_QE_DISPLAY_GetVersion	本モジュールのバージョンを読み出します。

1.3 制限事項

本モジュールは以下の制限事項があります。

- GLCDC FIT モジュールは Rev1.60 以上、SCI FIT モジュールは Rev5.20 以上を使用してください。
- 色深度の設定が 1～8 ビットの場合、CLUT テーブルは本モジュールや emWin、Aeropoint GUI FIT モジュールで使用するので、GLCDC FIT で設定しても上書きされます。
- emWin FIT モジュールや Aeropoint GUI FIT モジュールと組み合わせて使用する場合、表示画面が静止中のときに調整を行ってください。アニメーション中などダブルバッファリングが動作しているときには正しく調整することができません。

1.4 ソフトウェア構成

本モジュールのソフトウェア構成を図 1.1 に示します。

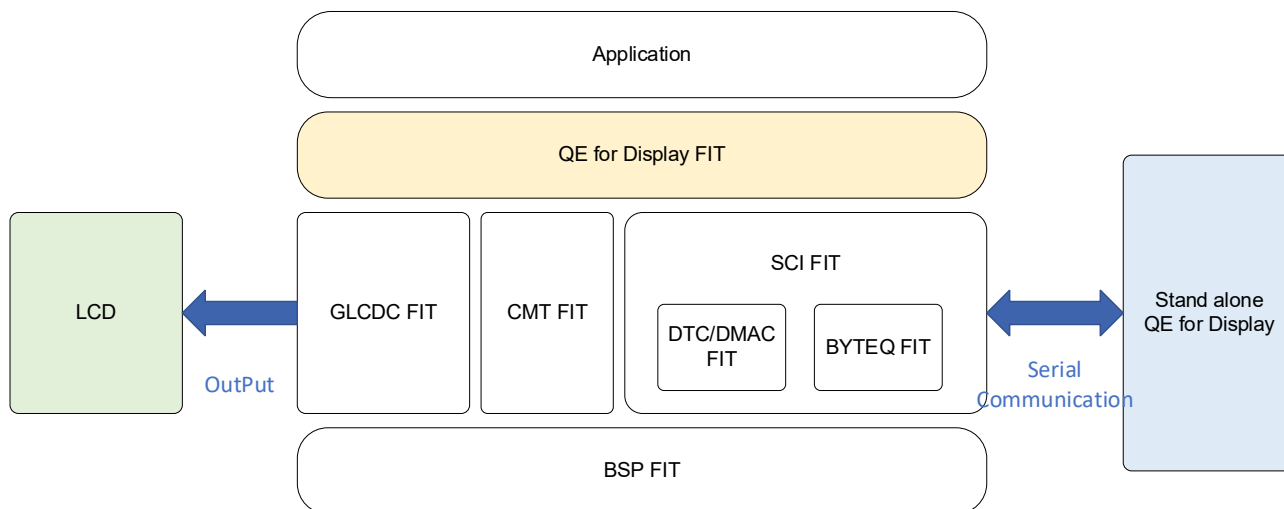


図 1.1 ソフトウェア構成

1.5 基本動作

本モジュールはシリアル通信を介して QE for Display からの接続要求やコマンドに対して応答します。PC と RX デバイスを接続して RX デバイスのプログラムを実行後、QE for Display からの接続を実施してください。

詳細な実行手順は QE for Display のマニュアルを参照してください。

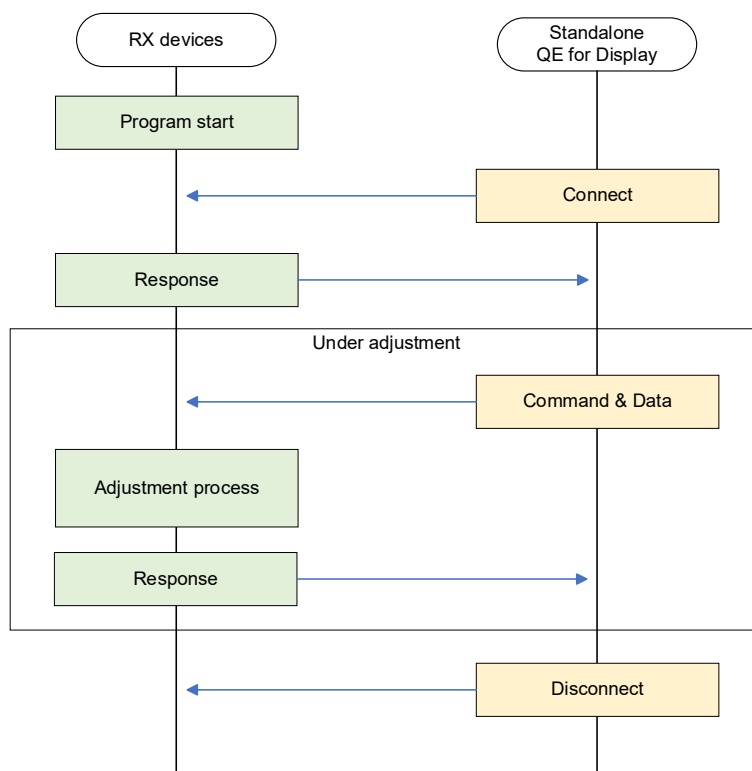


図 1.2 基本動作のイメージ

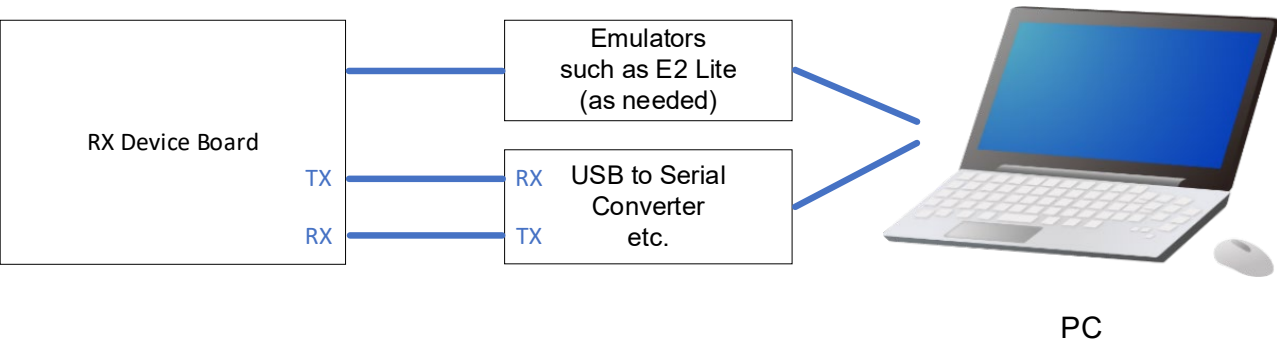


図 1.3 PC との接続

表 1.2 通信設定

項目	設定内容
シリアル通信方式	調歩同期式 (UART)
ボーレート (bps)	115200bps (デフォルト) コンフィギュレーションオプションで変更することができます。 詳細は 2.6 コンパイル時の設定を参照してください。
データ長	8 ビット
パリティ	なし
ストップビット	1 ビット
フロー制御	なし

2. API 情報

本 FIT モジュールは、下記の条件で動作を確認しています。

2.1 ハードウェアの要求

ご使用になる MCU が以下の機能をサポートしている必要があります。

- グラフィック LCD コントローラ(GLCDC)
- シリアルコミュニケーションインタフェース(SCI)
- コンペアマッチタイマ(CMT)

2.2 ソフトウェアの要求

このソフトウェアは以下の FIT モジュールに依存しています。

- ボードサポートパッケージ (r_bsp) v7.42 以上
- グラフィック LCD コントローラモジュール (r_glcdc_rx) v1.60 以上
- SCI モジュール (r_sci_rx) v5.20 以上
- CMT モジュール (r_cmt_rx)
- GPIO モジュール (r_gpio_rx)

2.3 サポートされているツールチェーン

本 FIT モジュールは「4.3 動作確認環境」に示すツールチェーンで動作確認を行っています。

2.4 ヘッダファイル

すべての API 呼び出しとそれをサポートするインタフェース定義は r_qe_display_rx_if.h に記載しています。

2.5 整数型

このドライバは ANSI C99 を使用しています。これらの型は stdint.h で定義されています。

2.6 コンパイル時の設定

本モジュールのコンフィギュレーションオプションの設定は、`r_qe_display_rx_config.h`で行います。
オプション名および設定値に関する説明を、下表に示します。

Configuration options in <code>r_qe_display_rx_config.h</code>	
QE_DISPLAY_CFG_UART_CH ※デフォルト値は “1”	QE for Display との通信に使用する SCI のチャネル番号を設定します。 設定範囲はご使用のデバイスに依存します。
QE_DISPLAY_CFG_UART_BAUDRATE ※デフォルト値は “115200”	QE for Display との通信に使用する SCI のボーレートを設定します。 QE for Display の設定値と合わせてください。 設定範囲はご使用のデバイスに依存します。
QE_DISPLAY_CFG_UART_INTERRUPT_PRIORITY_LEVEL ※デフォルト値は “5”	QE for Display との通信に使用する SCI の割り込み優先レベルを設定します。 設定範囲：0 ～ 15
QE_DISPLAY_CFG_DEBUG_PRINT_ENABLE ※デフォルト値は “0”	標準出力先（通常ではデバッガ接続中の開発環境のコンソール）に通信や調整のエラーメッセージを送信するかどうかを選択します。 “0”：エラーメッセージの送信は行いません。 “1”：エラーメッセージの送信を行います。 printf 関数を使用していますので、標準出力が使用できない環境（デバッガ未接続の状態など）では“0”を設定してください。 また、QE for Display との通信に影響を及ぼす場合がありますので、通常は“0”を設定してください。

また、本モジュールを使用する際、ご使用の開発環境においてプロジェクトの設定からマクロ

“**QE_DISPLAY_CONFIGURATION**” を定義してください。

このマクロが定義されていない状態でビルドした場合、本モジュールからエラーを出力します。

emWin FIT モジュールや Aeropoint GUI FIT モジュールと組み合わせる場合、併せて以下のマクロも定義してください。

emWin : “QE_EMWIN_CONFIGURATION”

Aeropoint GUI : “QE_AEROPOINT_CONFIGURATION”

2.7 コードサイズ

本モジュールの ROM サイズ、RAM サイズ、最大使用スタックサイズを下表に示します。代表して RX72N を掲載しています。

ROM (コードおよび定数) と RAM (グローバルデータ) のサイズは、ビルド時の「2.6 コンパイル時の設定」のコンフィギュレーションオプションによって決まります。

下表の値は下記条件で確認しています。

モジュールリビジョン: r_qe_display_rx rev1.10

コンパイラバージョン: Renesas Electronics C/C++ Compiler Package for RX Family V3.06.00

(統合開発環境のデフォルト設定に"-lang = c99"オプションを追加)

GCC for Renesas RX 8.3.0.202405

(統合開発環境のデフォルト設定に"-std=gnu99"オプションを追加)

IAR C/C++ Compiler for Renesas RX version 5.10.1

(統合開発環境のデフォルト設定)

コンフィギュレーションオプション: デフォルト設定

ROM、RAM およびスタックのコードサイズ				
デバイス	分類	使用メモリ		
		Renesas Compiler	GCC	IAR Compiler
RX72N	ROM	7075 バイト	9438 バイト	9953 バイト
	RAM	3284 バイト	3273 バイト	3140 バイト
	スタック	276 バイト	244 バイト	360 バイト

2.8 戻り値

API 関数の戻り値を示します。この列挙型は、API 関数のプロトタイプ宣言とともに `r_qe_display_rx_if.h` で記載されています。

```
/* 戻り値 */
typedef enum e_qe_display_err
{
    QE_DISPLAY_SUCCESS = 0,           // 正常終了
    QE_DISPLAY_ERR_NOT_OPEN,          // Open 関数が未実行
    QE_DISPLAY_ERR_ALREADY_OPENED,    // Open 関数が既に実行済み
    QE_DISPLAY_ERR_LCD_CONFIG,        // LCD の設定エラー
    QE_DISPLAY_ERR_LCD_OTHER,         // LCD のその他のエラー
    QE_DISPLAY_ERR_UART_CONFIG,       // シリアル通信の設定エラー
    QE_DISPLAY_ERR_UART_OTHER,        // シリアル通信のその他のエラー
} qe_display_err_t;
```

2.9 FIT モジュールの追加方法

本モジュールは、使用するプロジェクトごとに追加する必要があります。ルネサスでは、スマート・コンフィグレータを使用した(1)、(3)の追加方法を推奨しています。ただし、スマート・コンフィグレータは、一部の RX デバイスのみサポートしています。サポートされていない RX デバイスについては(2)の方法を使用してください。

- (1) CS+上でスマート・コンフィグレータを使用して FIT モジュールを追加する場合
CS+上で、スタンドアロン版スマート・コンフィグレータを使用して、自動的にユーザプロジェクトに FIT モジュールを追加します。詳細は、アプリケーションノート「RX スマート・コンフィグレータ ユーザーガイド: CS+編 (R20AN0470)」を参照してください。
- (2) CS+上で FIT モジュールを追加する場合
CS+上で、手動でユーザプロジェクトに FIT モジュールを追加します。詳細は、アプリケーションノート「RX ファミリ CS+に組み込む方法 Firmware Integration Technology (R01AN1826)」を参照してください。
- (3) IAREW 上でスマート・コンフィグレータを使用して FIT モジュールを追加する場合
スタンドアロン版スマート・コンフィグレータを使用して、自動的にユーザプロジェクトに FIT モジュールを追加します。詳細は、アプリケーションノート「RX スマート・コンフィグレータ ユーザーガイド: IAREW 編 (R20AN0535)」を参照してください。

2.10 for 文、while 文、do while 文について

FIT モジュールでは、レジスタの反映待ち処理などで for 文、while 文、do while 文（ループ処理）を使用しています。これらループ処理には、「WAIT_LOOP」をキーワードとしたコメントを記述しています。そのため、ループ処理にユーザがフェイルセーフの処理を組み込む場合、「WAIT_LOOP」で該当の処理を検索できます。

「WAIT_LOOP」を記述している対象デバイス

- ・ RX651, RX65N グループ
- ・ RX66N グループ
- ・ RX72N グループ
- ・ RX72M グループ

以下に記述例を示します。

```
while 文の例 :
/* WAIT_LOOP */
while(0 == SYSTEM.OSCOVFSR.BIT.PLOVF)
{
    /* The delay period needed is to make sure that the PLL has stabilized. */
}

for 文の例 :
/* Initialize reference counters to 0. */
/* WAIT_LOOP */
for (i = 0; i < BSP_REG_PROTECT_TOTAL_ITEMS; i++)
{
    g_protect_counters[i] = 0;
}

do while 文の例 :
/* Reset completion waiting */
do
{
    reg = phy_read(ether_channel, PHY_REG_CONTROL);
    count++;
} while ((reg & PHY_CONTROL_RESET) && (count < ETHER_CFG_PHY_DELAY_RESET)); /* WAIT_LOOP */
```

3. API 関数

3.1 R_QE_DISPLAY_Open ()

この関数は、GLCDC の初期化、および QE Display とのシリアル通信を初期化する関数です。この関数は他の API 関数を使用する前に実行する必要があります。

Format

```
qe_display_err_t R_QE_DISPLAY_Open (void)
```

Parameters

なし

Return Values

<code>QE_DISPLAY_SUCCESS</code>	<i>/* 正常終了 */</i>
<code>QE_DISPLAY_ERR_ALREADY_OPENED</code>	<i>/* Open 関数が既に実行済み */</i>
<code>QE_DISPLAY_ERR_LCD_CONFIG</code>	<i>/* LCD の設定エラー */</i>
<code>QE_DISPLAY_ERR_UART_CONFIG</code>	<i>/* シリアル通信の設定エラー */</i>

Properties

`r_qe_display_rx_if.h` にプロトタイプ宣言されています。

Description

本関数を実行することで GLCDC の初期化、および QE Display とのシリアル通信を初期化します。

Example

4.1 サンプルコードを参照してください。

Special Notes:

emWin FIT モジュール、および Aeropoint GUI FIT モジュールでデザインした画像を元に調整したい場合、あらかじめそれらの初期化処理を実行してから本関数を実行してください。

SCI FIT モジュールで DMAC/DTC を使用する場合、本モジュール内では DMAC や DTC FIT モジュールの初期化関数（`R_DMACE_Init` 関数、`R_DTC_Open` 関数）を実行していません。そのため、あらかじめそれらの初期化関数（`R_DMACE_Init` 関数、`R_DTC_Open` 関数）を実行してから本関数を実行してください。

3.2 R_QE_DISPLAY_Exec ()

QE Display からの受信データをもとに GLCDC の調整処理やエラー処理を実行する関数です。

Format

```
qe_display_err_t R_QE_DISPLAY_Exec (void)
```

Parameters

なし

Return Values

<code>QE_DISPLAY_SUCCESS</code>	<i>/* 正常終了 */</i>
<code>QE_DISPLAY_ERR_NOT_OPEN</code>	<i>/* Open 関数が未実行 */</i>
<code>QE_DISPLAY_ERR_LCD_OTHER</code>	<i>/* LCD のその他のエラー */</i>
<code>QE_DISPLAY_ERR_UART_OTHER</code>	<i>/* シリアル通信のその他のエラー */</i>

Properties

r_qe_display_rx_if.h にプロトタイプ宣言されています。

Description

GLCDC の調整処理を実行します。本関数をメインルーチンで繰り返し実行されるようにしてください。

Example

4.1 サンプルコードを参照してください。

Special Notes:

なし

3.3 R_QE_DISPLAY_GetVersion ()

この関数は、API のバージョンを返す関数です。

Format

```
uint32_t R_QE_DISPLAY_GetVersion (void)
```

Parameters

なし

Return Values

バージョン番号

Properties

r_qe_display_rx_if.h にプロトタイプ宣言されています。

Description

現在インストールされている本 FIT モジュールのバージョンを返します。バージョン番号はコード化されています。最初の 2 バイトがメジャーバージョン番号で、後の 2 バイトがマイナーバージョン番号です。例えば、バージョンが 4.25 の場合、戻り値は '0x00040019' となります。

Example

```
/* FIT モジュールのバージョンを取得する場合 */  
volatile uint32_t  version;  
  
version = R_QE_DISPLAY_GetVersion();
```

Special Notes:

なし

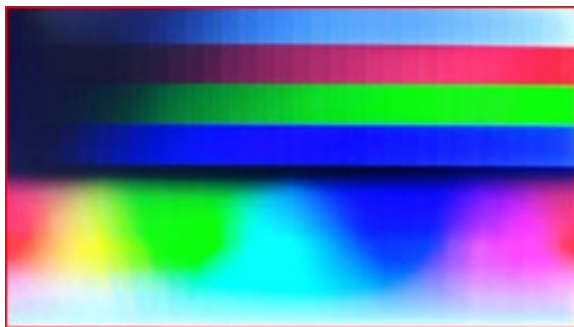
4. 付録

4.1 サンプルコード

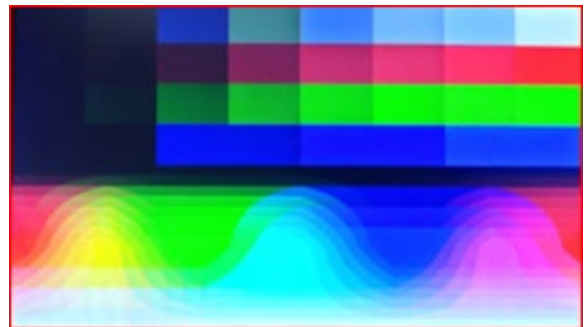
本モジュールのサンプルコードを下記に示します。GLCDC FIT モジュールのみ使用する場合は 4.1.1 を、emWin FIT モジュールと併せて使用する場合は 4.1.2 を、Aeropoint GUI FIT モジュールと併せて使用する場合は 4.1.3 を参照してください。

GLCDC FIT モジュールのみ使用する場合、実行後の画面には下記のデモ画面が表示されます。

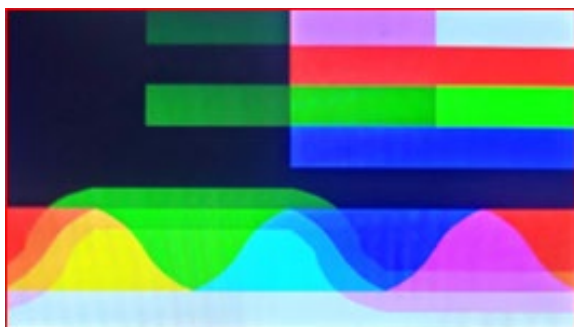
ARGB8888, RGB888, RGB565, ARGB1555, ARGB444



CLUT8



CLUT4



CLUT1

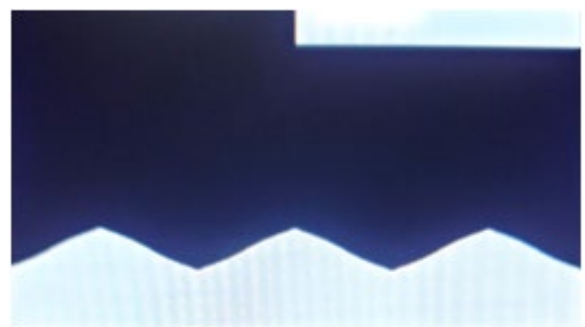


図 4.1 デモ画面

4.1.1 GLCDC FIT モジュールのみ使用する場合のサンプルコード

```
#include "r_smc_entry.h"
#include "r_qe_display_rx_if.h"
#include "r_glcddc_rx_if.h"

/* Initialize a first time interrupt flag
 * Unintended specified line notification from graphic 2 and
 * graphic 1, 2 underflow is detected only
 * for first time after release GLCDC software reset.
 * This variable is a flag to skip the first time interrupt processing.
 * Refer to Graphic LCD Controller (GLCDC) section of
 * User's Manual: Hardware for details. */
bool first_interrupt_flag = false;

void main(void)
{
    qe_display_err_t ret;

    /* When using DMAC/DTC with uart, initialize the DMAC/DTC
     * before using the R_QE_Display_Open function.*/
    //R_DMACA_Init(); or R_DTC_Open();

    ret = R_QE_DISPLAY_Open();
    if(ret != QE_DISPLAY_SUCCESS)
    {
        while(1)
        {
            /* Check GLCDC and SCI setting */
        }
    }

    while(1)
    {
        R_QE_DISPLAY_Exec();
    }
}

/* If you use the GLCDC callback function,
/* please enable the settings required for interrupt generation.
/* Ex. VPOS interrupt
/* - LCD_CH0_DETECT_VPOS (true)
/* - LCD_CH0_INTERRUPT_VPOS_ENABLE (true)
/* - LCD_CH0_CALLBACK_ENABLE (true)
/* - LCD_CH0_PCALLBACK (my_glcddc_callback)

void my_glcddc_callback(void * pdata)
{
    if (false == first_interrupt_flag)
    {
        first_interrupt_flag = true;
        /* do nothing */
    }
    else
    {
        glcddc_callback_args_t * pdecode;
```

```
pdecode = (glcdc_callback_args_t *)pdata;  
...  
}  
}
```


4.1.2 emWin FIT モジュールを使用する場合のサンプルコード

```
#include "r_smc_entry.h"
#include "r_qe_display_rx_if.h"

/* 1. API programming */
#include "GUI.h"

void main(void)
{
    qe_display_err_t ret;

    /* When using DMAC/DTC with uart, initialize the DMAC/DTC
     * before using the R_QE_Display_Open function.
     * When using DMAC, initialization is not required
     * if it is initialized within the emWin FIT module. */
    //R_DMACA_Init(); or R_DTC_Open();

    /* initialize emwin */
    GUI_Init();

    /* ~~ emWin drawing user code ~~ */

    ret = R_QE_DISPLAY_Open();
    if(ret != QE_DISPLAY_SUCCESS)
    {
        while(1)
        {
            /* Check GLCDC and SCI setting */
        }
    }

    while(1)
    {
        GUI_Exec();
        R_QE_DISPLAY_Exec();
    }
}

/* 2. Use AppWizard */
#include "Generated/Resource.h"

void main(void)
{
    qe_display_err_t ret;

    /* When using DMAC/DTC with uart, initialize the DMAC/DTC
     * before using the R_QE_Display_Open function.
     * When using DMAC, initialization is not required
     * if it is initialized within the emWin FIT module. */
    //R_DMACA_Init(); or R_DTC_Open();

    APPW_X_Setup();
    APPW_Init(APPW_PROJECT_PATH);
    APPW_CreatePersistentScreens();
    APPW_CreateRoot(APPW_INITIAL_SCREEN, WM_HBKWIN);

    ret = R_QE_DISPLAY_Open();
```

```
if(ret != QE_DISPLAY_SUCCESS)
{
    while(1)
    {
        /* Check GLCDC and SCI setting */
    }
}

while (1) {
    while (GUI_Exec1()) {
        APPW_Exec();
    }
    APPW_Exec();
    GUI_Delay(5);
    R_QE_DISPLAY_Exec();
}
}
```

4.1.3 Aeropoint GUI FIT モジュールを使用する場合のサンプルコード

```
#include <stdio.h>
#include "r_smc_entry.h"
#include "cri_aero_config_ais.h"
#include "cri_aero_player.h"
#include "cri_aero_error.h"
#include "rx72n_env_test_GUI.h"
#include "r_qe_display_rx_if.h"

void main(void)
{
    qe_display_err_t ret;

    CriBool flag = CRI_TRUE;

    /* When using DMAC/DTC with uart, initialize the DMAC/DTC
     * before using the R_QE_Display_Open function.
     * When using DMAC, initialization is not required
     * if it is initialized within the Aeropoint GUI FIT module. */
    //R_DMACA_Init(); or R_DTC_Open();

    /* Initialize */
    #if CRI_AERO_CONFIG_AIS_USE_SD || CRI_AERO_CONFIG_AIS_USE_FLASH
    /* When using slides in SD or external FLASH. */
    flag = CriAeroPlayer_Initialize(NULL);
    #else
    /* When using slides in Memory. */
    flag = CriAeroPlayer_Initialize( &rx72n_env_test_GUI_gData );
    #endif

    if(flag == CRI_FALSE)
    {
        CriUint32 errorNo = CriAeroError_GetLastError();
        printf("CriAeroPlayer_Initialize failed. errorNo:%d", errorNo);
        for(;;)
        {
        }
    }

    #if 1
    /* Switch from title to slide 1. */
    /* You can also switch slides by LAN or UART command. */
    CriAeroPlayer_ReadSlide( 1 );
    #endif

    ret = R_QE_DISPLAY_Open();
    if(QE_DISPLAY_SUCCESS != ret)
    {
        while(1)
        {
            /* Check GLCDC and SCI setting */
        }
    }

    /* Main loop */
    while (flag)
    {
        /* Main */
    }
}
```

```
    flag = CriAeroPlayer_Main();  
    R_QE_DISPLAY_Exec() ;  
}  
  
/* Finalize */  
CriAeroPlayer_Finalize();  
  
return;  
}
```

4.2 開発環境 (CS+、EWRX)の設定

4.2.1 マクロ定義の設定

本モジュールを使用する場合に各開発環境でマクロ定義 “QE_DISPLAY_CONFIGURATION” の設定が必要です。マクロ定義 “QE_DISPLAY_CONFIGURATION” の設定方法を示します。

emWin FIT モジュールと組み合わせる場合は “QE_EMWIN_CONFIGURATION” を、Aerpoint GUI FIT モジュールと組み合わせる場合は “QE_AERPOINT_CONFIGURATION” を併せて定義してください。

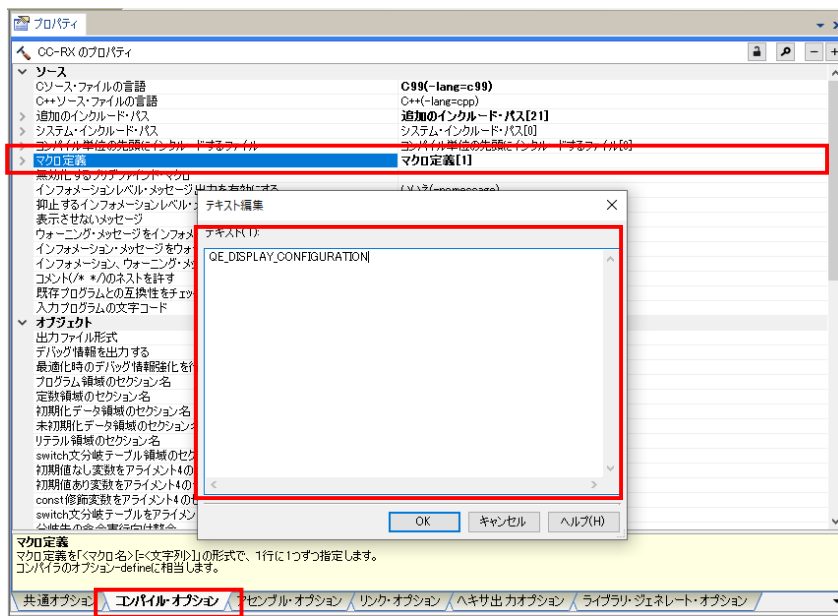


図 4.2 CS+の設定

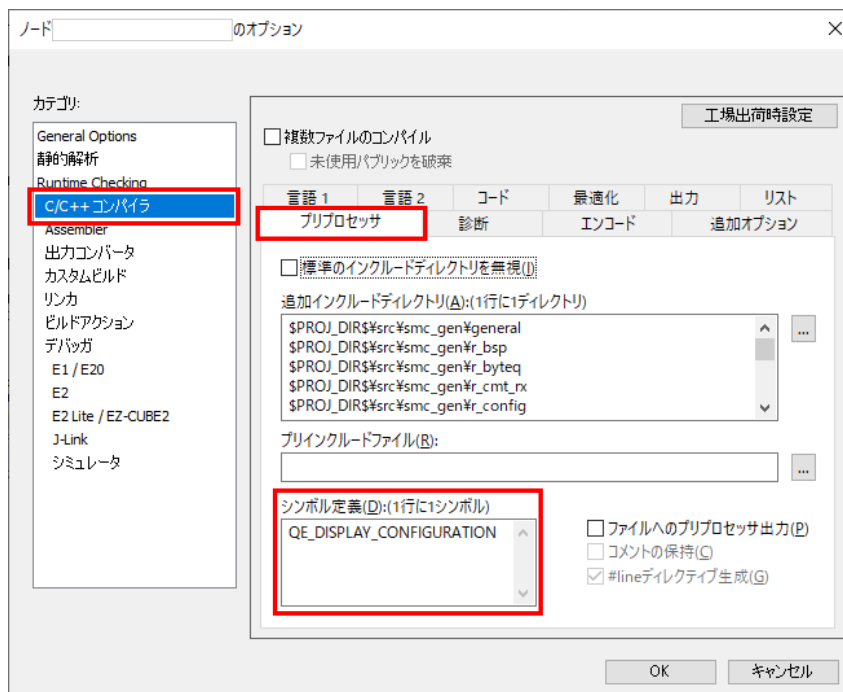


図 4.3 EWRX の設定

4.2.2 デバッグ・コンソールの表示方法

本モジュールは調整や通信エラー発生時に標準出力先にエラーメッセージを送信します。以下に CS+と EWRX のコンソール画面の表示方法を示します。

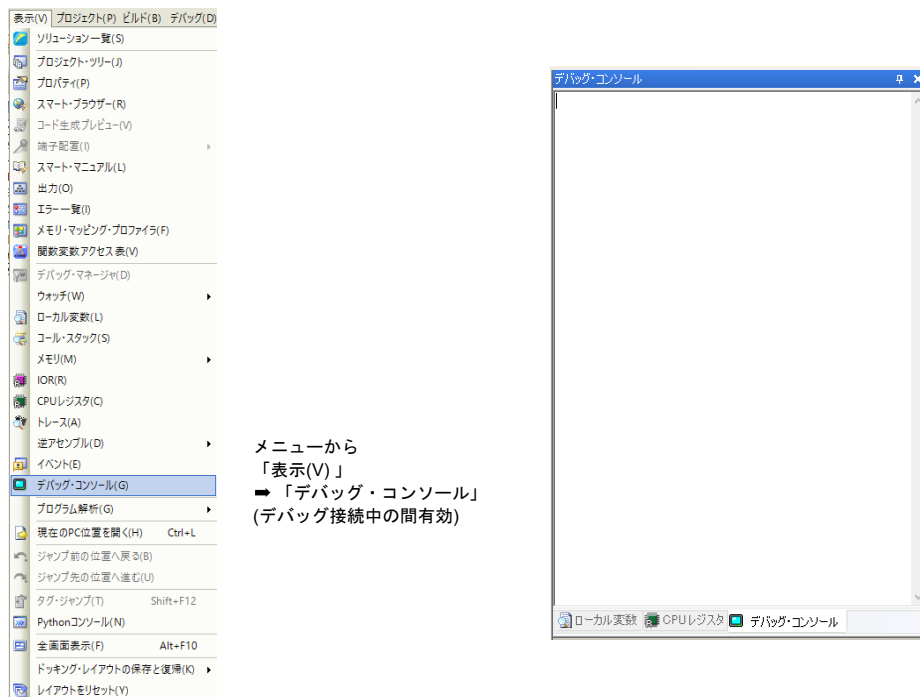


図 4.4 CS+の設定

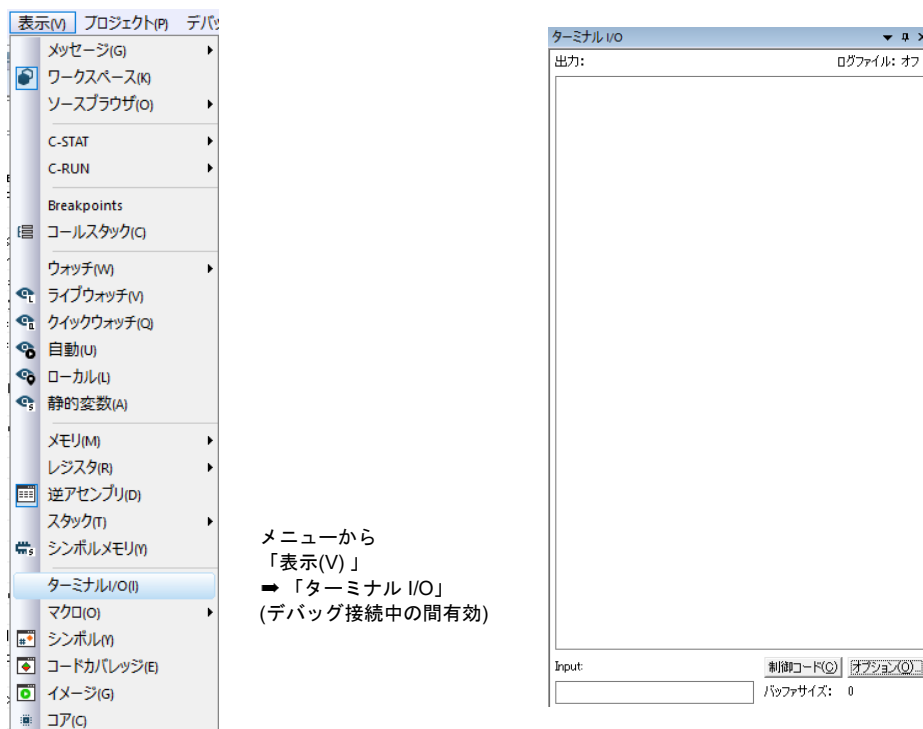


図 4.5 EWRX の設定

4.3 動作確認環境

本 FIT モジュールの動作確認環境を以下に示します。

表 4.1 動作確認環境 (Rev.1.00)

項目	内容
統合開発環境	ルネサスエレクトロニクス製 e ² studio 2024-04 IAR Embedded Workbench for Renesas RX 5.10.1
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler for RX Family V3.06.00 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -lang = c99
	GCC for Renesas RX 8.03.00.202405 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -std=gnu99
	IAR C/C++ Compiler for Renesas RX version 5.10.01 コンパイルオプション：統合開発環境のデフォルト設定
エンディアン	ビッグエンディアン/リトルエンディアン
モジュールのバージョン	Rev.1.00
使用ボード	Renesas Envision Kit RPBRX65N (型名：RTK5RX65N2C00000BR) Renesas Envision Kit RPBRX72N (型名：RTK5RX72N0C00000BJ) Renesas Starter Kit+ for RX72N (型名：RTK5572NNHS10000BE)

表 4.2 動作確認環境 (Rev.1.10)

項目	内容
統合開発環境	ルネサスエレクトロニクス製 e ² studio 2024-07 IAR Embedded Workbench for Renesas RX 5.10.1
C コンパイラ	ルネサスエレクトロニクス製 C/C++ Compiler for RX Family V3.06.00 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -lang = c99
	GCC for Renesas RX 8.03.00.202405 コンパイルオプション：統合開発環境のデフォルト設定に以下のオプションを追加 -std=gnu99
	IAR C/C++ Compiler for Renesas RX version 5.10.01 コンパイルオプション：統合開発環境のデフォルト設定
エンディアン	ビッグエンディアン/リトルエンディアン
モジュールのバージョン	Rev.1.10
使用ボード	Renesas Envision Kit RPBRX65N (型名：RTK5RX65N2C00000BR) Renesas Envision Kit RPBRX72N (型名：RTK5RX72N0C00000BJ) Renesas Starter Kit+ for RX72N (型名：RTK5572NNHS10000BE)

。

4.4 トラブルシューティング

- (1) Q : 本 FIT モジュールをプロジェクトに追加しましたが、ビルド実行すると「Error!! Please declare QE_DISPLAY_CONFIGURATION definition to the compiler.」または「Could not open source file "r_image_config.h"」、「Could not open source file "r_lcd_timing.h"」が発生します。

A : スタンドアロン版 QE for Display からファイルが出力されていること、プロジェクトにマクロ定義“QE_DISPLAY_CONFIGURATION”が設定されていることを確認してください。

- (2) Q : QE for Display との通信が正常に開始できずエラーが発生します。

A : SCI FIT モジュールの設定や使用端子、配線などに問題がないことや、ボーレートを遅くして通信できるかどうかを確認してください。ボーレートは本モジュールのコンフィギュレーションオプションで設定します。

- (3) Q : 調整実行中に調整を行っても LCD に何も反映されません。

A : 設定項目によっては LCD の表示に表れないことがあります。まずは輝度やコントラストの画質調整をお試しいただき、LCD に反映されることを確認してください。QE for Display からの反応や LCD への反映が全く見られない場合、QE for Display との接続解除後、デバイスをリセットしてから再度調整をお試しください。また、コンフィギュレーションオプションの QE_DISPLAY_CFG_DEBUG_PRINT_ENABLE を“1”に設定すると、標準出力先（CS+でデバッグしている場合は「デバッグ・コンソール」）にエラーメッセージを出力していますので、メッセージの出力内容を確認してください。

5. 参考ドキュメント

ユーザーズマニュアル：ハードウェア

（最新版をルネサス エレクトロニクスホームページから入手してください。）

テクニカルアップデート／テクニカルニュース

（最新の情報をルネサス エレクトロニクスホームページから入手してください。）

ユーザーズマニュアル：開発環境

RX ファミリ CC-RX コンパイラ ユーザーズマニュアル（R20UT3248）

（最新版をルネサス エレクトロニクスホームページから入手してください。）

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	Jul. 19. 24		新規作成
1.10	Sep. 20. 24	8	2.7 コードサイズ 修正
		23	4.3 動作確認環境 表 4.2 追加
		プログラム	1. SCI FIT モジュールの受信バッファサイズ (SCI_CFG_CHx_RX_BUFSIZ) をデフォルト値 “80” から 変更すると通信エラーが発生する問題を修正。 2.QE for Display の TCON と LCD 設定タブにおいて「出力 データのビットエンディアン」、「出力データのピクセル順 序」が正しく設定されない問題を修正。

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うと、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。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、お客様の責任において、お客様の機器・システムを設計ください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品または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. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 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。
6. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。

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

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

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

7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア／ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限られません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因したまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア／ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
8. 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。
9. 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。
10. 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。
11. 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。
12. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものいたします。
13. 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。
14. 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.5.0-1 2020.10)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

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

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。