

## RXファミリ

### emWin v6.22モジュール Firmware Integration Technology

---

#### 要旨

本アプリケーションノートは、Firmware Integration Technology (FIT)を使用した emWin v6.22 モジュールについて説明します。以降、本モジュールを emWin FIT モジュールと称します。

emWin FIT モジュールは SEGGER 社の emWin (<https://www.segger.com/products/user-interface/emwin/add-ons/emwin-support-renesas-rx-mcu/>) を FIT モジュール化したものです。

「emWin」および GUI デザインツール「AppWizard」の詳細に関しては、SEGGER 社 (<https://www.segger.com/>) または国内代理店のエンビテック社 (<https://www.embitek.co.jp/>) にお問合せください。

#### 動作確認デバイス

- ・ RX65N グループ、RX651 グループ      ROM 容量 : 1.5MB ~ 2MB
- ・ RX72N グループ                      ROM 容量 : 1.5MB ~ 4MB

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あわせて変更し、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8.1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	4
1.1 emWin FITモジュールとは	4
1.2 emWin FITモジュールの概要	4
1.3 製品構成	5
1.4 APIの概要	7
1.5 ソフトウェア構成	9
2. API情報	10
2.1 ハードウェアの要求	10
2.2 ソフトウェアの要求	10
2.3 サポートされているツールチェーン	10
2.4 ヘッダファイル	10
2.5 整数型	10
2.6 コンパイル時の設定	10
2.7 コードサイズ	12
2.8 引数	14
2.9 FITモジュールの追加方法	14
2.10 for文、while文、do while文について	15
3. emWinから呼び出される関数	16
3.1 GUI_X_Config()	16
3.2 LCD_X_Config ()	17
3.3 LCD_X_DisplayDriver ()	18
3.4 GUI_X_Init ()	20
3.5 GUI_X_Delay ()	21
3.6 GUI_X_ExecIdle ()	22
3.7 GUI_X_GetTime ()	23
3.8 GUI_X_ErrorOut ()	24
3.9 GUI_X_Warn ()	25
3.10 GUI_X_Log ()	26
3.11 GUI_X_InitOS ()	27
3.12 GUI_X_Unlock ()	28
3.13 GUI_X_Lock ()	29
3.14 GUI_X_GetTaskId ()	30
3.15 GUI_X_WaitEvent ()	31
3.16 GUI_X_SignalEvent ()	32
3.17 GUI_X_WaitEventTimed ()	33
3.18 PID_X_SetLayerIndex ()	34
3.19 PID_X_Init ()	35
3.20 GUI_TOUCH_X_ActiveX ()	36
3.21 GUI_TOUCH_X_ActiveY ()	37
3.22 GUI_TOUCH_X_MeasureX ()	38
3.23 GUI_TOUCH_Y_MeasureY ()	39
3.24 APPW_X_FS_Init ()	40

4.	アプリケーションから呼び出されるAPI関数 .....	41
4.1	R_EMWIN_GetBufferAddr() .....	41
4.2	R_EMWIN_GetD2 ().....	42
4.3	R_EMWIN_EnableDave2D ().....	43
4.4	R_EMWIN_DisableDave2D ().....	44
4.5	R_EMWIN_GetDaveActive ().....	45
4.6	R_EMWIN_GetVersion ().....	46
4.7	_VSYNC_ISR ().....	47
5.	端子設定 .....	48
6.	実装上の注意事項 .....	49
6.1	ライブラリファイルの選択 .....	49
6.2	動作確認環境以外での使用 .....	49
7.	サンプルアプリケーション .....	51
8.	付録 .....	52
8.1	動作確認環境 .....	52
8.2	トラブルシューティング .....	52
9.	参考ドキュメント .....	54
	テクニカルアップデートの対応について .....	54
	改訂記録 .....	55

## 1. 概要

### 1.1 emWin FIT モジュールとは

emWin FIT モジュールは API として、プロジェクトに組み込んで使用します。emWin FIT モジュールの組み込み方については、「2.9FITモジュールの追加方法」を参照してください。

### 1.2 emWin FIT モジュールの概要

emWin FIT モジュールは、SEGGER 社の emWin (v6.22) を FIT 化することで、スマート・コンフィグレータにより簡単にユーザプログラムへ emWin を組み込めるようにしたものです。今後は、v6.22 以降のバージョンアップにも対応して行く予定です。

emWin の詳細については以下のドキュメントを参照してください。

- ・ emWin Graphic Library with Graphical User Interface User Guide & Reference Manual

(<https://www.segger.com/downloads/emwin/UM03001>)

emWin FIT モジュールには、以下の制限事項があります。

- ・ DRW2D FIT モジュールの使用を推奨
- ・ OS は FreeRTOS、OS レスのみ対応
- ・ SEGGER 社製の emFILE および embOS には非対応

emWin FIT モジュール使用時には以下の点に注意してください。

- ・ RX65N を使用する場合にセクション設定が必要

emWin FIT モジュールでは、256Kbyte のバッファを 2 つ確保する必要があります。RX65N で使用する場合には、アドレス配置の関係より、バッファを 2 ヶ所に分けて配置する必要があります。そのため、バッファを 0x00000100 から 256Kbyte、0x00800000 から 256Kbyte を確保した場合、元から設定されている SU 以降のセクションは、0x00840000 以降に設定してください。

- ・ ヒープサイズを初期設定値から変更が必要

スマート・コンフィグレータを使用し「r\_bsp」の「Heap size」を 0x4000 に変更してください。

- ・ コンパイラオプションの設定が必要

コンパイラオプションの“const 修飾変数をアライメント数が 4 のセクションに配置する(-nostuff=C)”の設定が必要です。

e<sup>2</sup> studio の場合、[プロジェクト]→[C/C++ Project Settings]でプロパティ画面を開き、[C/C++ビルド]→[設定]の[ツール設定]タブを開き、[Compiler]→[オブジェクト]の“const 修飾変数をアライメント数が 4 のセクションに配置する(-nostuff=C)”の項目にチェックを入れてください。

- ・ 画像の形式

emWin FIT モジュールで画像等を使用する場合は、ビットマップ形式(.bmp)のデータを使用してください。

## 1.3 製品構成

本製品は、以下の表 1-2のファイルが含まれます。

表 1-1 製品構成

ファイル/ディレクトリ(太字)名	内容
r01an5983jj0100-rx-emwin.pdf	emWin FIT モジュール アプリケーションノート(日本語)
r01an5983ej0100-rx-emwin.pdf	emWin FIT モジュール アプリケーションノート(英語)
<b>FITModules</b>	FIT モジュールフォルダ
r_emwin_rx_v6.22_100.mdf	emWin FIT モジュール スマート・コンフィグレータ用コンフィグレーション設定ファイル
r_emwin_rx_v6.22_100.xml	emWin FIT モジュール e2 studio FIT プラグイン用 XML ファイル
r_emwin_rx_v6.22_100.zip	emWin FIT モジュール (以下は展開時のフォルダ内容物)
<b>r_config</b>	emWin FIT モジュール用のコンフィグ H ファイル格納フォルダ
r_emwin_rx_config.h	emWin FIT モジュール用のコンフィグ H ファイル
<b>r_emwin_rx</b>	emWin FIT モジュールのソースコードとドキュメント及び関連ツール格納用フォルダ
readme.txt	emWin FIT モジュールの概要とファイル構成についての説明
r_emwin_rx_if.h	emWin FIT モジュール用の宣言 H ファイル
<b>doc</b>	emWin FIT モジュールのドキュメント格納フォルダ
<b>emWin_doc</b>	SEGGER 社提供の、emWin ライブラリに関するドキュメント格納用フォルダ
<b>en</b>	emWin FIT モジュール アプリケーションノート(英語)格納用フォルダ
r01an5983ej0100-rx-emwin.pdf	emWin FIT モジュール アプリケーションノート(英語)
<b>ja</b>	emWin FIT モジュール アプリケーションノート(日本語)格納用フォルダ
r01an5983jj0100-rx-emwin.pdf	emWin FIT モジュール アプリケーションノート(日本語)
<b>Training</b>	SEGGER 社提供の、emWin ライブラリを使用するサンプルプログラム (詳細は、本フォルダに格納されている emWin_Training.pdf を参照してください。)

<b>lib</b>	emWin ライブラリ、及びこれに対するコンフィグレーションとして用意しているインタフェース部のソースコードとヘッダファイル格納用フォルダ
<b>Config</b>	emWin ライブラリに対するインタフェース部のソースコードとヘッダファイル格納用フォルダ
APPW_X_NoFS.c	AppWizard を使用するための関数を含む C ファイル (※ファイルシステムを使用しない場合の関数のみサポートしています)
GUI_X_Ex.c	RTOS の使用時、及び非使用時に使用する関数を含む C ファイル
GUIConf.c	emWin ライブラリが使用するワークメモリの確保と初期化に使用する関数を含む C ファイル
GUIConf.h	emWin ライブラリのコンフィグレーションを示す H ファイル
LCDConf.c	emWin ライブラリのディスプレイドライバの使用と初期化、及び H/W によるアクセラレータを担う関数を含む C ファイル
LCDConf.h	ディスプレイドライバの使用に関する H ファイル
PIDConf.c	タッチ機能の使用と初期化を担う関数を含む C ファイル
PIDConf.h	タッチ機能の使用に関する H ファイル
<b>GUI</b>	emWin ライブラリのライブラリファイルとヘッダファイル格納用フォルダ
<b>ref</b>	r_emwin_rx_config.h を手動で作成際に参照するヘッダファイル格納用フォルダ
r_emwin_rx_config_reference.h	r_emwin_rx_config.h を手動で作成際に参照するヘッダファイル
<b>src</b>	emWin ライブラリに対するインタフェース部以外のソースコードとヘッダファイル格納用フォルダ
r_emwin_rx_if.c	emWin FIT モジュール固有の関数を含む C ファイル
r_emwin_rx_lcdif.c	LCD とのインタフェース機能を担う C ファイル
r_emwin_rx_pidif.c	タッチパネルとのインタフェース機能を担う C ファイル
r_emwin_rx_private.h	emWin FIT モジュール内部で使用する H ファイル
<b>tool</b>	AppWizard インストーラを始めとする emWin ライブラリを使用するためのツール類格納フォルダ (詳細は、doc/Training フォルダに格納されている emWin_Training.pdf を参照してください。)

## 1.4 API の概要

emWin FIT モジュールに含まれる API 関数を表 1.2に示します。詳細は3章emWinから呼び出される関数、4章アプリケーションから呼び出されるAPI関数を参照してください。

表 1.2 emWin が内部で呼び出す関数一覧

関数	関数説明
GUI_X_Config	emWin のメモリ管理システムで使用するメモリブロックの登録に使用します。
LCD_X_Config	LCD とデバイスドライバの初期化に使用します。
LCD_X_DisplayDriver	ディスプレイドライバのコールバック関数として使用します。
GUI_X_Init	必要なハードウェアを初期化します。
GUI_X_Delay	指定の時間を待ちます。
GUI_X_ExecIdle	Window Manager から、GUI が最新の状態で処理すべき内容がないときに呼び出されます。
GUI_X_GetTime	現在のシステム時間がミリ秒単位の整数型で得られます。
GUI_X_ErrorOut	致命的なエラーが発生した際に、エラー文字列を入力として emWin から呼び出されます。
GUI_X_Warn	警告が発生した際に、警告文字列を入力として emWin から呼び出されます。
GUI_X_Log	メッセージが発生した際に、メッセージ文字列を入力として emWin から呼び出されます。
GUI_X_InitOS	マルチタスク環境で使用する場合に、セマフォまたは mutex を生成します。
GUI_X_Unlock	マルチタスク環境で使用する場合に、GUI をアンロックします。
GUI_X_Lock	マルチタスク環境で使用する場合に、GUI をロックします。
GUI_X_GetTaskId	マルチタスク環境で使用する場合に、タスク ID を取得します。
GUI_X_WaitEvent	マルチタスク環境で使用する場合に、イベント待ちを実施します。
GUI_X_SignalEvent	マルチタスク環境で使用する場合に、イベント通知を実施します。
GUI_X_WaitEventTimed	マルチタスク環境で使用する場合に、指定した期間のイベント待ちを実施します。
PID_X_SetLayerIndex	レイヤー番号を設定します。
PID_X_Init	Pointer Input Device を初期化します。
GUI_TOUCH_X_ActiveX	Touch IC の X 軸の電圧測定を有効にします。
GUI_TOUCH_X_ActiveY	Touch IC の Y 軸の電圧測定を有効にします。
GUI_TOUCH_X_MeasureX	Touch IC から得られる X 軸の電圧測定結果を返します。
GUI_TOUCH_Y_MeasureY	Touch IC から得られる Y 軸の電圧測定結果を返します。
APPW_X_FS_Init	AppWizard のファイルシステムアクセスを初期化します。

表 1.3 アプリケーションから呼び出される API 関数一覧

関数	関数説明
R_EMWIN_GetBufferAddr	フレームバッファのアドレスを取得します。
R_EMWIN_GetD2	Dave2D 機能のハンドルを取得します。
R_EMWIN_EnableDave2D	Dave2D 機能を動作許可状態にします。
R_EMWIN_DisableDave2D	Dave2D 機能を動作禁止状態にします。
R_EMWIN_GetDaveActive	Dave2D 機能の動作状態を取得します。
R_EMWIN_GetVersion	emWin のバージョンを取得します。
_VSYNC_ISR()	Vsync 割り込み処理を行います。(GLCDC のコールバック関数を想定)



## 1.5 ソフトウェア構成

emWin FIT モジュールを使用したアプリケーションは、図 1.1のようなソフトウェア構成になります。

アプリケーションからは emWin FIT モジュールを使用します。emWin FIT モジュールから DRW2D FIT モジュールを使用して図形を生成し、GLCDC FIT モジュールを使用して LCD への表示を行い、タッチパネル情報は SCI-I2C FIT モジュールを使用して制御します。

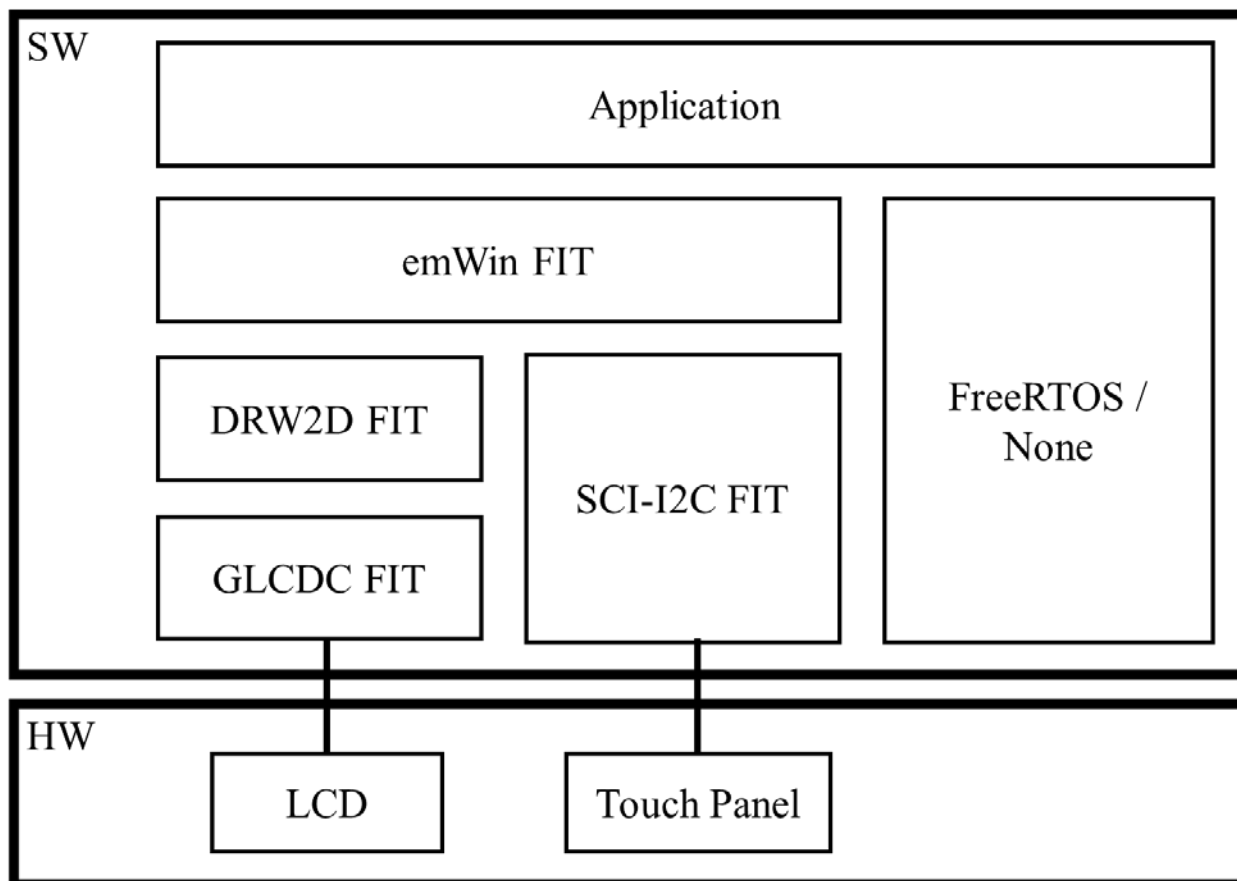


図 1.1 ソフトウェア構成

## 2. API 情報

本 FIT モジュールを使用して実装する際には、下記の事項に注意してください。

### 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。

- GPIO
- DMAC
- CMT
- SCI
- GLCDC

### 2.2 ソフトウェアの要求

このドライバは以下の 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) Rev.6.20 以降
- GPIO (r\_gpio\_rx) Rev.4.10 以降
- DMAC (r\_dmaca\_rx) Rev.2.60 以降
- CMT (r\_cmt\_rx) Rev.4.90 以降
- SCI(簡易 I<sup>2</sup>C モード) (r\_sci\_iic\_rx) Rev.2.49 以降
- グラフィック LCD コントローラ (r\_glcdc\_rx) Rev.1.50 以降
- DRW2D (r\_drw2d\_rx) Rev.1.11 以降

### 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「8.1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\_emwin\_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

emWin FIT 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\_emwin\_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 r_emwin_rx_config.h	
EMWIN_GUI_NUM_BYTES	GUI で使用する最大メモリサイズを設定してください。
EMWIN_BITS_PER_PIXEL	ピクセル当たりの色深度を選択してください。 “1” の場合、色深度 1bpp が設定されます。この場合、DRW2D を使用する

	<p>ことはできません。</p> <p>“4”の場合、色深度 4bpp が設定されます。この場合、DRW2D を使用することはできません。</p> <p>“8”の場合、色深度 8bpp が設定されます。この場合、DRW2D を使用することはできません。</p> <p>“16”の場合、色深度 16bpp が設定されます。この値が推奨値となります。</p> <p>“32”の場合、色深度 32bpp が設定されます。</p>
EMWIN_NUM_BUFFERS	<p>画像の描画に使用するフレームバッファ数を設定してください。emWin FIT モジュールのサポートしている範囲は 1~3 です。それ以上のフレームバッファを使用する場合は、フレームバッファ数設定箇所の実装を修正してください。</p>
EMWIN_GUI_FRAME_BUFFER1	<p>画像の描画に使用するフレームバッファ 1 の先頭アドレスを設定してください。</p>
EMWIN_GUI_FRAME_BUFFER2	<p>画像の描画に使用するフレームバッファ 2 の先頭アドレスを設定してください。</p>
EMWIN_GUI_FRAME_BUFFER3	<p>画像の描画に使用するフレームバッファ 3 の先頭アドレスを設定してください。</p>
EMWIN_DMACH_NUMBER	<p>フレームバッファ間のデータ転送に使用する DMACH のチャンネル番号を設定してください。</p>
EMWIN_INIT_DMACH	<p>emWin FIT モジュール内で DMACH の初期化を行うかを設定してください。</p> <p>“0”の場合、emWin FIT モジュール内で DMACH の初期化を行いません。</p> <p>“1”の場合、emWin FIT モジュール内で DMACH の初期化を行います。</p>
EMWIN_USE_DRW2D	<p>DRW2D 使用の有無を選択してください。</p> <p>“0”の場合、DRW2D を使用しません。</p> <p>“1”の場合、DRW2D を使用します。</p>
EMWIN_XSIZE_PHYS	<p>横方向の表示画面サイズを設定してください。</p>
EMWIN_YSIZE_PHYS	<p>縦方向の表示画面サイズを設定してください。</p>
EMWIN_DISPLAY_ORIENTATION	<p>LCD に画像を表示する向きを定義値から選択してください。</p> <p>“ORIENTATION_0”の場合、画面に正対する向きに表示します。</p> <p>“ORIENTATION_CW”の場合、時計回りに 90 度回転した向きに表示します、</p> <p>“ORIENTATION_180”の場合、180 度回転した向きに表示します。</p> <p>“ORIENTATION_CCW”の場合、反時計回りに 90 度回転した向きに表示します。</p>
EMWIN_DISPLAY_DRIVER	<p>emWin ライブラリが内部で使用するディスプレイドライバを定義値から選択してください。</p> <p>“GUIDRV_LIN”の場合、GUIDRV_Lin を使用します。</p> <p>“GUIDRV_OTHER”の場合、使用するディスプレイドライバを実装してください。</p>
EMWIN_LCD_IF	<p>LCD に画像を表示するためのインターフェースを選択してください。</p> <p>“LCD_IF_GLCDC”の場合、GLCDC を使用します。</p> <p>“LCD_IF_OTHER”の場合、使用するインターフェースを実装してください。</p>
EMWIN_USE_DISP_SIGNAL_PIN	<p>LCD のリセット制御ピン使用の有無を選択してください。</p> <p>“0”の場合、LCD のリセット制御ピンを使用しません。</p> <p>“1”の場合、LCD のリセット制御ピンを使用します。</p>
EMWIN_DISP_SIGNAL_PIN	<p>LCD のリセット制御ピン使用時に、使用する GPIO のピン番号を設定してください。設定には、GPIO FIT モジュールの列挙型 gpio_port_pin_t のメンバを使用してください。</p>
EMWIN_USE_BACKLIGHT_PIN	<p>LCD のバックライト制御ピン使用の有無を選択してください。</p> <p>“0”の場合、LCD のバックライト制御ピンを使用しません。</p> <p>“1”の場合、LCD のバックライト制御ピンを使用します。</p>
EMWIN_BACKLIGHT_PIN	<p>LCD のバックライト制御ピン使用時に、使用する GPIO のピン番号を設定してください。設定には、GPIO FIT モジュールの列挙型 gpio_port_pin_t のメンバを使用してください。</p>
EMWIN_USE_TOUCH	<p>タッチ機能使用の有無を選択してください。</p>

	<p>“0” の場合、タッチ機能を使用しません。</p> <p>“1” の場合、タッチ機能を使用します。</p>
EMWIN_TOUCH_IF	<p>タッチ機能で使用するインタフェースを定義値から選択してください。</p> <p>“TOUCH_IF_SCI_IIC” の場合、SCI-I<sup>2</sup>C を使用します。</p> <p>“TOUCH_IF_OTHER” の場合、使用するインタフェースを実装してください。</p>
EMWIN_TOUCH_IF_NUMBER	<p>タッチ機能で使用するインタフェースのチャンネル番号を設定してください。</p>
EMWIN_USE_MULTITOUCH	<p>マルチタッチ機能の使用の有無を選択してください。</p> <p>“0” の場合、マルチタッチ機能を使用しません。</p> <p>“1” の場合、マルチタッチ機能を使用します。</p>
EMWIN_MAX_NUM_TOUCHPOINTS	<p>マルチタッチ機能使用時に、タッチパネルのポイントの最大数を設定してください。</p>
EMWIN_SLAVE_ADDRESS	<p>タッチパネルのスレーブアドレスを設定してください。</p>
EMWIN_USE_TOUCH_IC_RESET_PIN	<p>タッチパネル IC リセットピン使用の有無を選択してください。</p> <p>“0” の場合、タッチパネル IC リセットピンを使用しません。</p> <p>“1” の場合、タッチパネル IC リセットピンを使用します。</p>
EMWIN_TOUCH_IC_RESET_PIN	<p>タッチパネル IC リセットピン使用時に、使用する GPIO のピン番号を設定してください。設定には、GPIO FIT モジュールの列挙型 gpio_port_pin_t のメンバを使用してください。</p>

## 2.7 コードサイズ

emWin FIT 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。

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: emWin Rev6.22 FIT Rev.1.00

コンパイラバージョン: Renesas Electronics C/C++ Compiler Package for RX Family V3.03.00

(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 8.03.00.202004

(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 4.20.1

(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
RX65N	ROM	103K バイト	127K バイト	59K バイト
	RAM	84K バイト	84K バイト	83K バイト
	スタック	408 バイト	440 バイト	576 バイト
RX72N	ROM	118K バイト	128K バイト	59K バイト

	RAM	84K バイト	84K バイト	83K バイト
	スタック	620 バイト	440 バイト	576 バイト

---

## 2.8 引数

---

API 関数の引数である構造体を示します。この構造体は、API 関数のプロトタイプ宣言とともに `r_emwin_rx_if.h` に記載されています。

---

## 2.9 FIT モジュールの追加方法

---

emWin FIT 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)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) e<sup>2</sup> studio 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

e<sup>2</sup> studio のスマート・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: e<sup>2</sup> studio 編 (R20AN0451)」を参照してください。

※ダウンロードした FIT モジュールの保存先に、他のバージョンの emWin FIT モジュールが存在する場合、適切にモジュールを追加することができないことがあります。該当のフォルダ内には最新版の emWin FIT モジュールのみを格納してください。

(2) e<sup>2</sup> studio 上で FIT 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

e<sup>2</sup> studio の FIT 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 モジュールを追加することができます。詳細は、アプリケーションノート「RX ファミリ e<sup>2</sup> studio に組み込む方法 Firmware Integration Technology (R01AN1723)」を参照してください。

## 2.10 for 文、while 文、do while 文について

本モジュールでは、レジスタの反映待ち処理等で 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は、「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
- ・RX72M グループ
- ・RX72N 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. emWin から呼び出される関数

---

#### 3.1 GUI\_X\_Config()

---

この関数は、emWin のメモリ管理システムで使用するメモリブロックを登録する関数です。

**Format**

void GUI\_X\_Config(void)

**Parameters**

無し

**Return Values**

無し

**Properties**

GUI.h にプロトタイプ宣言されています。

**Description**

emWin のメモリ管理システムで使用するメモリブロックの登録に使用されます。

emWin FIT モジュールでは、GUI ブロックの関数を使用してメモリを割り当てています、

**Reentrant**

- 不可



---

## 3.2 LCD\_X\_Config ()

---

この関数は、LCD とデバイスドライバを初期化する関数です。

### Format

void LCD\_X\_Config(void)

### Parameters

無し

### Return Values

無し

### Properties

LCD.h にプロトタイプ宣言されています。

### Description

LCD とデバイスドライバの初期化に使用されます。

emWin FIT モジュールでは、GUI ブロックの関数を使用して LCD を初期化し、あわせて DRW2D FIT モジュールを初期化しています。

### Reentrant

- 不可

---

### 3.3 LCD\_X\_DisplayDriver ()

---

この関数は、ディスプレイドライバのコールバック関数です。

#### Format

```
int LCD_X_DisplayDriver(  
    unsigned layer_index,  
    unsigned cmd,  
    void * p_data  
)
```

#### Parameters

layer_index	入力	レイヤー番号
cmd	入力	実行するコマンド
p_data	入力	データ構造へのポインタ

#### Return Values

0:	コマンドが正常に実行された
-1:	コマンドが実行されなかった
-2:	エラー発生

#### Properties

LCD.h にプロトタイプ宣言されています。

#### Description

ディスプレイドライバのコールバック関数として使用されます。ディスプレイドライバから呼び出され、コールバックルーチンを実行します。

emWin FIT モジュールでは、コマンドに応じて GLCDC FIT モジュールの初期化と DRW2D FIT モジュールを使用した図形生成関数の登録、Lookup Table エントリの設定、ディスプレイの ON/OFF、バッファの切り替えを行っています。

コマンド	値	意味	対応状況 ○ : 対応 × : 未対応
LCD_X_INITCONTROLLER	0x01	ディスプレイコントローラの初期化	○
LCD_X_SETVRAMADDR	0x02	Video RAM アドレスの設定	×
LCD_X_SETORG	0x03	レイヤー内の基準の設定	×
LCD_X_SETLUTENTRY	0x04	Lookup Table のエントリの設定	○
LCD_X_ON	0x05	ディスプレイのスイッチオン	○
LCD_X_OFF	0x06	ディスプレイのスイッチオフ	○
LCD_X_SETSIZE	0x07	レイヤーサイズの設定	×
LCD_X_SETPOS	0x08	レイヤー位置の設定	×
LCD_X_SETVIS	0x09	レイヤーの可視化の設定	×
LCD_X_SETALPHA	0x0A	レイヤーのアルファ値の設定	×
LCD_X_SETALPHAMODE	0x0B	アルファブレンディングモードの設定	×
LCD_X_SETCHROMAMODE	0x0C	クロマブレンディングモードの設定	×
LCD_X_SETCHROMA	0x0D	クロマ値の設定	×
LCD_X_SHOWBUFFER	0x0E	バッファの切り替え	○

**Reentrant**

- 不可

---

### 3.4 GUI\_X\_Init ()

---

この関数は、GUIに必要なハードウェアを初期化する関数です。

#### Format

void GUI\_X\_Init(void)

#### Parameters

無し

#### Return Values

無し

#### Properties

GUI.h にプロトタイプ宣言されています。

#### Description

必要なハードウェアを初期化する関数です。

emWin FIT モジュールでは、待ち時間の計測に使用するためのコンペアマッチタイマの初期化に使用されています。

#### Reentrant

- 不可

---

### 3.5 GUI\_X\_Delay ()

---

この関数は、指定の時間を待つ関数です。

#### Format

```
void GUI_X_Delay(  
    int ms  
)
```

#### Parameters

ms 入力                    待ち時間[ミリ秒]

#### Return Values

無し

#### Properties

GUI.h にプロトタイプ宣言されています。

#### Description

指定の時間を待ちます。

emWin FIT モジュールでは、コンペアマッチタイマから得られる時間情報を利用して、指定した時間を待ちます。

#### Reentrant

- 不可

### 3.6 GUI\_X\_ExecIdle ()

---

この関数は、Window Manager から GUI が最新の状態のため処理すべき内容がないときに呼び出される関数です。

**Format**

void GUI\_X\_ExecIdle(void)

**Parameters**

無し

**Return Values**

無し

**Properties**

GUI.h にプロトタイプ宣言されています。

**Description**

Window Manager から、GUI が最新の状態で処理すべき内容がないときに呼び出されます。

emWin FIT モジュールでは、何も処理をしません。

**Reentrant**

- 不可

---

### 3.7 GUI\_X\_GetTime ()

---

この関数は、現在のシステム時間がミリ秒単位の整数型で得られる関数です。

#### Format

```
GUI_TIMER_TIME GUI_X_GetTime(  
    int ms  
)
```

#### Parameters

無し

#### Return Values

システム時間[ミリ秒]

#### Properties

GUI.h にプロトタイプ宣言されています。

#### Description

現在のシステム時間がミリ秒単位の整数型で得られます。

emWin FIT モジュールでは、コンペアマッチタイマから取得される値を返しています。

#### Reentrant

- 不可

---

## 3.8 GUI\_X\_ErrorOut ()

---

この関数は、致命的なエラーが発生した際に、エラー文字列を入力として emWin から呼び出される関数です。

### Format

```
void GUI_X_ErrorOut(  
    const char *s  
)
```

### Parameters

s    入力                    エラー文字列

### Return Values

無し

### Properties

GUI.h にプロトタイプ宣言されています。

### Description

致命的なエラーが発生した際に、エラー文字列を入力として emWin から呼び出されます。

GUI\_DEBUG\_LEVEL  $\geq$  3 で有効となります。

emWin FIT モジュールでは、何も処理をしません。

### Reentrant

- 不可



---

### 3.9 GUI\_X\_Warn ()

---

この関数は、警告が発生した際に、警告文字列を入力として emWin から呼び出される関数です。

#### Format

```
void GUI_X_Warn(  
    const char *s  
)
```

#### Parameters

s 入力 警告文字列

#### Return Values

無し

#### Properties

GUI.h にプロトタイプ宣言されています。

#### Description

警告が発生した際に、警告文字列を入力として emWin から呼び出されます。

GUI\_DEBUG\_LEVEL  $\geq$  4 で有効となります。

emWin FIT モジュールでは、何も処理をしません。

#### Reentrant

- 不可

---

### 3.10 GUI\_X\_Log ()

---

この関数は、メッセージが発生した際に、メッセージ文字列を入力として emWin から呼び出される関数です。

#### Format

```
void GUI_X_Log(  
    const char *s  
)
```

#### Parameters

s 入力                   メッセージ文字列

#### Return Values

無し

#### Properties

GUI.h にプロトタイプ宣言されています。

#### Description

メッセージが発生した際に、メッセージ文字列を入力として emWin から呼び出されます。

GUI\_DEBUG\_LEVEL  $\geq$  5 で有効となります。

emWin FIT モジュールでは、何も処理をしません。

#### Reentrant

- 不可

---

### 3.11 GUI\_X\_InitOS ()

---

この関数は、マルチタスク環境で使用する場合に、セマフォまたは mutex を生成する関数です。

#### Format

void GUI\_X\_InitOS(void)

#### Parameters

無し

#### Return Values

無し

#### Properties

GUI.h にプロトタイプ宣言されています。

#### Description

マルチタスク環境で使用する場合に、セマフォまたは mutex を生成する関数です。

emWin FIT モジュールでは、FreeRTOS 使用時は、FreeRTOS の関数を使用して、セマフォの生成とイベントの生成を行っています。FreeRTOS 不使用時は、何も処理をしません。

#### Reentrant

- 不可

---

## 3.12 GUI\_X\_Unlock ()

---

この関数は、マルチタスク環境で使用する場合に、GUI をアンロックする関数です。

### Format

void GUI\_X\_Unlock(void)

### Parameters

無し

### Return Values

無し

### Properties

GUI.h にプロトタイプ宣言されています。

### Description

マルチタスク環境で使用する場合に、GUI をアンロックする関数です。

emWin FIT モジュールでは、FreeRTOS 使用時は、FreeRTOS の関数を使用して、セマフォを解放しています。FreeRTOS 不使用時は、何も処理をしません。

### Reentrant

- 不可

---

### 3.13 GUI\_X\_Lock ()

---

この関数は、マルチタスク環境で使用する場合に、GUI をロックする関数です。

#### Format

void GUI\_X\_Unlock(void)

#### Parameters

無し

#### Return Values

無し

#### Properties

GUI.h にプロトタイプ宣言されています。

#### Description

マルチタスク環境で使用する場合に、GUI をロックする関数です。

emWin FIT モジュールでは、FreeRTOS 使用時は、FreeRTOS の関数を使用して、セマフォを取得しています。FreeRTOS 不使用時は、何も処理をしません。

#### Reentrant

- 不可

---

### 3.14 GUI\_X\_GetTaskId ()

---

この関数は、マルチタスク環境で使用する場合に、タスク ID を取得する関数です。

#### Format

U32 GUI\_X\_GetTaskId(void)

#### Parameters

無し

#### Return Values

タスク ID

#### Properties

GUI.h にプロトタイプ宣言されています。

#### Description

マルチタスク環境で使用する場合に、タスク ID を取得する関数です。

emWin FIT モジュールでは、FreeRTOS 使用時は、FreeRTOS の関数を使用して、タスク ID を取得しています。FreeRTOS 不使用時は、固定で 1 を返しています。

#### Reentrant

- 不可

---

### 3.15 GUI\_X\_WaitEvent ()

---

この関数は、マルチタスク環境で使用する場合に、イベント待ちを実施する関数です。

#### Format

void GUI\_X\_WaitEvent(void)

#### Parameters

無し

#### Return Values

無し

#### Properties

GUI.hにプロトタイプ宣言されています。

#### Description

マルチタスク環境で使用する場合に、イベント待ちを実施する関数です。

emWin FIT モジュールでは、FreeRTOS 使用時は、FreeRTOS の関数を使用してイベント待ちを行っています。ここで、最大待ち時間は60000 ミリ秒としています。FreeRTOS 不使用時は、何も処理をしません。

#### Reentrant

- 不可

---

### 3.16 GUI\_X\_SignalEvent ()

---

この関数は、マルチタスク環境で使用する場合に、イベント通知を実施する関数です。

#### Format

void GUI\_X\_SignalEvent(void)

#### Parameters

無し

#### Return Values

無し

#### Properties

GUI.hにプロトタイプ宣言されています。

#### Description

マルチタスク環境で使用する場合に、イベント通知を実施する関数です。

emWin FIT モジュールでは、FreeRTOS 使用時は、FreeRTOS の関数を使用してイベントを通知しています。FreeRTOS 不使用时は、何も処理をしません。

#### Reentrant

- 不可



---

### 3.17 GUI\_X\_WaitEventTimed ()

---

この関数は、マルチタスク環境で使用する場合に、指定した期間のイベント待ちを実施する関数です。

#### Format

```
void GUI_X_WaitEventTimed(  
    int period  
)
```

#### Parameters

Period	入力	指定期間
--------	----	------

#### Return Values

無し

#### Properties

GUI.h にプロトタイプ宣言されています。

#### Description

マルチタスク環境で使用する場合に、指定した期間のイベント待ちを実施する関数です。

emWin FIT モジュールでは、FreeRTOS 使用時は、FreeRTOS の関数を使用して、指定した期間のイベント待ちを行っています。ここで、最大待ち時間は 60000 ミリ秒としています。FreeRTOS 不使用时は、何も処理をしません。

#### Reentrant

- 不可

---

### 3.18 PID\_X\_SetLayerIndex ()

---

この関数は、レイヤー番号を設定する関数です。

#### Format

```
void PID_X_SetLayerIndex(  
    int layer_index  
)
```

#### Parameters

LayerIndex 入力                      レイヤー番号

#### Return Values

無し

#### Properties

PIDConf.h にプロトタイプ宣言されています。

#### Description

レイヤー番号を設定します。

emWin FIT モジュールでは、内部変数にレイヤー番号を設定します。

#### Reentrant

- 不可

---

### 3.19 PID\_X\_Init ()

---

この関数は、Pointer Input Device を初期化する関数です。

#### Format

void PID\_X\_Init(void)

#### Parameters

無し

#### Return Values

無し

#### Properties

PIDConf.h にプロトタイプ宣言されています。

#### Description

Pointer Input Device の初期化を行います。

emWin FIT モジュールでは、Touch IC のリセット、SCI-I2C の初期化、タッチ情報を取得するためのコンペアマッチタイマ起動とコールバック関数登録、マルチタッチ機能の有効化を行っています。

#### Reentrant

- 不可

---

### 3.20 GUI\_TOUCH\_X\_ActiveX ()

---

この関数は、Touch IC の X 軸の電圧測定を有効にする関数です。

**Format**

```
void GUI_TOUCH_X_ActivateX(void)
```

**Parameters**

無し

**Return Values**

無し

**Properties**

GUI.h にプロトタイプ宣言されています。

**Description**

Touch IC の X 軸の電圧測定を有効にする関数です。

emWin FIT モジュールでは、何も処理をしません。

**Reentrant**

- 不可

---

### 3.21 GUI\_TOUCH\_X\_ActiveY ()

---

この関数は、Touch IC の Y 軸の電圧測定を有効にする関数です。

**Format**

void GUI\_TOUCH\_X\_ActivateY(void)

**Parameters**

無し

**Return Values**

無し

**Properties**

GUI.h にプロトタイプ宣言されています。

**Description**

Touch IC の Y 軸の電圧測定を有効にする関数です。

emWin FIT モジュールでは、何も処理をしません。

**Reentrant**

- 不可

---

## 3.22 GUI\_TOUCH\_X\_MeasureX ()

---

この関数は、Touch IC から得られる X 軸の電圧測定結果を返す関数です。

### Format

```
int GUI_TOUCH_X_MeasureX(void)
```

### Parameters

無し

### Return Values

0

### Properties

GUI.h にプロトタイプ宣言されています。

### Description

Touch IC から得られる X 軸の電圧測定結果を返す関数です。

emWin FIT モジュールでは、固定で 0 を返します。

### Reentrant

- 不可

---

### 3.23 GUI\_TOUCH\_Y\_MeasureY ()

---

この関数は、Touch IC から得られる Y 軸の電圧測定結果を返す関数です。

#### Format

```
int GUI_TOUCH_X_MeasureY(void)
```

#### Parameters

無し

#### Return Values

0

#### Properties

GUI.h にプロトタイプ宣言されています。

#### Description

Touch IC から得られる Y 軸の電圧測定結果を返す関数です。

emWin FIT モジュールでは、固定で 0 を返します。

#### Reentrant

- 不可

---

### 3.24 APPW\_X\_FS\_Init ()

---

この関数は、AppWizard のファイルシステムアクセスを初期化する関数です。

#### Format

void APPW\_X\_FS\_Init (void)

#### Parameters

無し

#### Return Values

無し

#### Properties

AppWizard.h にプロトタイプ宣言されています。

#### Description

AppWizard のファイルシステムアクセスを初期化する関数です。

emWin FIT モジュールでは、何も処理をしません。

#### Reentrant

- 不可



## 4. アプリケーションから呼び出される API 関数

---

### 4.1 R\_EMWIN\_GetBufferAddr()

---

この関数は、emWin FIT モジュールで使用されるフレームバッファのアドレスを取得する関数です。

**Format**

void \* R\_EMWIN\_GetBufferAddr (void)

**Parameters**

無し

**Return Values**

フレームバッファ・アドレス

**Properties**

r\_emwin\_rx\_if.h にプロトタイプ宣言されています。

**Description**

emWin FIT モジュールで使用されるフレームバッファのアドレスを取得します。

**Reentrant**

- 不可

---

## 4.2 R\_EMWIN\_GetD2 ()

---

この関数は、emWin FIT モジュールの Dave2D 機能のハンドルを取得する関数です。

### Format

d2\_device \* R\_EMWIN\_GetD2 (void)

### Parameters

無し

### Return Values

Dave2D のハンドル

### Properties

r\_emwin\_rx\_if.h にプロトタイプ宣言されています。

### Description

emWin FIT モジュールの Dave2D 機能のハンドルを取得します。

本関数は、DRW2D FIT モジュール使用時のみ有効になります。

### Reentrant

- 不可

---

### 4.3 R\_EMWIN\_EnableDave2D ()

---

この関数は、emWin FIT モジュールの Dave2D 機能を動作許可状態にする関数です。

#### Format

void R\_EMWIN\_EnableDave2D (void)

#### Parameters

無し

#### Return Values

無し

#### Properties

r\_emwin\_rx\_if.h にプロトタイプ宣言されています。

#### Description

emWin FIT モジュールの Dave2D 機能を動作許可状態にします。

本関数は、DRW2D FIT モジュール使用時のみ有効になります。

#### Reentrant

- 不可

---

## 4.4 R\_EMWIN\_DisableDave2D ()

---

この関数は、emWin FIT モジュールの Dave2D 機能を動作禁止状態にする関数です。

### Format

void R\_EMWIN\_DisableDave2D (void)

### Parameters

無し

### Return Values

無し

### Properties

r\_emwin\_rx\_if.h にプロトタイプ宣言されています。

### Description

emWin FIT モジュールの Dave2D 機能を動作禁止状態にします。

本関数は、DRW2D FIT モジュール使用時のみ有効になります。

### Reentrant

- 不可

---

## 4.5 R\_EMWIN\_GetDaveActive ()

---

この関数は、emWin FIT モジュールの Dave2D 機能の動作状態を取得する関数です。

### Format

uint32\_t R\_EMWIN\_GetDaveActive (void)

### Parameters

無し

### Return Values

Dave2D 動作状態 (0:動作禁止状態、1:動作許可状態)

### Properties

r\_emwin\_rx\_if.h にプロトタイプ宣言されています。

### Description

emWin FIT モジュールの Dave2D 機能の動作状態を取得します。

本関数は、DRW2D FIT モジュール使用時のみ有効になります。

### Reentrant

- 不可

---

## 4.6 R\_EMWIN\_GetVersion ()

---

この関数は、emWin FIT モジュールのバージョン番号を取得する関数です。

### Format

```
void R_EMWIN_GetVersion(st_emwin_version_t * version)
```

### Parameters

\* version    出力    バージョン番号の格納先ポインタ

### Return Values

無し

### Properties

r\_emwin\_rx\_if.h にプロトタイプ宣言されています。

### Description

emWin FIT モジュールのバージョン番号を取得します。

### Reentrant

- 不可

---

## 4.7 \_VSYNC\_ISR ()

---

この関数は、V-sync 割り込みの処理を行う関数です。

### Format

void \_VSYNC\_ISR(void \* p)

### Parameters

\* p            出力            GLCDC からの Callback argument

### Return Values

無し

### Properties

r\_emwin\_rx\_if.h にプロトタイプ宣言されています。

### Description

V-sync 割り込みの処理を行います。

GLCDC FIT モジュールのコールバック関数を想定しています。

### Reentrant

- 不可

## 5. 端子設定

emWin FIT モジュールを使用するための端子設定を、QE for Display[RX]で行うことができます。

設定が必要な端子は、LCD パネルのリセット端子、LCD パネルのバックライト端子、LCD パネル搭載のタッチ IC のリセット端子です。

e<sup>2</sup> studio の場合は、QE for Display[RX]の emWin の設定ダイアログの端子設定機能を使用することで端子設定を行うことができます。QE for Display[RX]を使用する場合、r\_emwin\_rx に関するスマート・コンフィグレータによる端子の設定は不要です。

選択した端子情報は qe\_emwin\_config.h に出力され、2.6コンパイル時の設定に示すマクロ定義の値が設定されます。その際に、r\_emwin\_rx\_config.h のマクロ定義は無効化されます。

QE for Display[RX]を使用せず端子設定を行う場合は、emWin FIT モジュールに含まれる r\_emwin\_rx\_config\_reference.h を編集し、r\_emwin\_rx\_config.h を作成してください。



## 6. 実装上の注意事項

本 FIT モジュールを使用して実装する際には、下記の事項に注意してください。

### 6.1 ライブラリファイルの選択

emWin FIT モジュールには、以下のライブラリファイルが含まれています。使用する MCU とコンパイラに応じたライブラリファイルを選択して使用してください。

ライブラリファイル	
emWinLib_CCRX.lib	Renesas Electronics C/C++ Compiler for RX Family で使用するライブラリファイル。
libemWinLib_GCC.a	GCC for Renesas RX で使用するライブラリファイル。
emWinLib_RXv2_IAR.a	RX65N 等の RXv2 コアを搭載した MCU を使用するプロジェクトにおいて IAR C/C++ Compiler for Renesas RX で使用するライブラリファイル。
emWinLib_RXv3_IAR.a	RX72N 等の RXv3 コアを搭載した MCU を使用するプロジェクトにおいて IAR C/C++ Compiler for Renesas RX で使用するライブラリファイル。

### 6.2 動作確認環境以外での使用

emWin FIT モジュールを、動作確認環境に示されている環境以外で使用する際には、以下の事項に注意して使用してください。動作確認環境については、8.1章を参照してください。

#### ・LCD の使用

emWin FIT モジュールでは、GLCDC FIT モジュールを使用して LCD への描画を行っています。LCD とのインタフェースの設定には、以下の方法を使用することができます。

##### 1. QE for Display を使用した設定

e<sup>2</sup> studio においてスマート・コンフィグレータを使用して emWin FIT モジュールを追加した後に、QE for Display において必要な設定を入力してください。詳細については、以下の URL 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](#)

##### 2. スマート・コンフィグレータによる設定

プロジェクト作成時にスマート・コンフィグレータを使用し、ソフトウェアコンポーネント設定画面において必要な設定を入力してください。

※QE for Display を使用した場合、その設定によりスマート・コンフィグレータで設定した項目は上書きされます。

##### 3. 設定データ構造体の実装

スマート・コンフィグレータ/QE for Display を使用せず、r\_emwin\_rx\_lcdconf.c ファイル内の r\_emwin\_lcd\_open 関数において、R\_GLCDC\_Open 関数を使用している箇所を設定データ構造体を実装してください。

また、GLCDC FIT モジュールを使用しない場合は、r\_emwin\_rx\_lcdconf.c 内の GLCDC FIT モジュールを使用して実施している処理を代替する処理を実装してください。e<sup>2</sup> studio を使用して実装する場合、r\_emwin\_rx\_config.h の EMWIN\_LCD\_IF に LCD\_IF\_OTHER を設定することで、LCD とのインタフェースを実装する必要のある箇所に warning が表示されます。

#### ・タッチパネルの使用

emWin FIT モジュールでは、動作確認環境で示されている評価ボードに実装されているタッチパネルを I<sup>2</sup>C インタフェースで使用する場合の処理が実装されています。他のタッチパネル、他のインタフェースを使用する場合には、以下の部分に必要な処理を実装してください。

##### 1. PIDConf.c ファイル内の、タッチデータを取得し emWin ライブラリに渡している部分

##### 2. r\_emwin\_rx\_pidconf.c 内の、タッチパネルとのインタフェースを担っている部分

2については、e<sup>2</sup> studio を使用して実装する場合、r\_emwin\_rx\_config.h の EMWIN\_TOUCH\_IF に

TOUCH\_IF\_OTHER を設定することで、タッチパネルとのインタフェースを実装する必要のある箇所に warning が表示されます。

#### ・ OS の使用

emWin FIT モジュールでは、FreeRTOS を使用する場合 (BSP\_CFG\_RTOS\_USED == 1) 、または OS を使用しない場合 (BSP\_CFG\_RTOS\_USED == 0) をサポートしています。これ以外の OS を使用する場合には、GUI\_X\_Ex.c 内の OS システムコールを使用している部分に、必要な処理を実装してください。

e<sup>2</sup> studio を使用して実装する場合、BSP\_CFG\_RTOS\_USED が上記の値以外であれば、OS システムコールを実装する必要のある箇所に warning が表示されます。詳細は、以下のドキュメントを参照してください。

ボードサポートパッケージモジュール Firmware Integration Technology (R01AN1685)

#### ・ emWin ライブラリの設定

emWin FIT モジュールでは、emWin ライブラリへの設定の内、フレームバッファ数については 3 面まで、ディスプレイドライバとしては GUIDRV\_Lin をサポートしています。表示のちらつきを抑えるためにフレームバッファ数を増やす場合、また上記以外のディスプレイドライバを適用する場合には、以下の部分に必要な処理を実装してください。

1. フレームバッファ数について、4 面以上を使用する場合には、r\_emwin\_rx\_config.h の EMWIN\_NUM\_BUFFERS に 16 を超えない範囲で使用するバッファ数を指定して、必要な実装を追加してください。e<sup>2</sup> studio を使用して実装する場合、EMWIN\_NUM\_BUFFERS に 4 以上の値を設定することで、LCDConf.c 内の追加実装が必要な箇所に warning が表示されます。
2. ディスプレイドライバについて、GUIDRV\_Lin 以外のディスプレイドライバを使用する場合には、LCDConf.c 内のディスプレイドライバを指定する部分の実装を修正してください。e<sup>2</sup> studio を使用して実装する場合、r\_emwin\_rx\_config.h の EMWIN\_DISPLAY\_DRIVER に GUIDRV\_OTHER を設定することで、LCDConf.c 内の修正が必要な箇所に warning が表示されます。

## 7. サンプルアプリケーション

doc/Training フォルダ内に、サンプルアプリケーションが格納されています。詳細については以下のドキュメントを参照してください。

- emWin Training

## 8. 付録

## 8.1 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 8.1 動作確認環境(Envision Kit)

項目	内容
統合開発環境	ルネサスエレクトロニクス製 e <sup>2</sup> studio 2021-10
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(CC-RX) V3.03.00 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 8.03.00.202004 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 4.20.1 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	リトルエンディアン
モジュールのバージョン	Ver.1.00
使用 OS	FreeRTOS <a href="#">Release Release RX MCUs FreeRTOS v1.0.1 comes from original 10.4.3 · renesas/FreeRTOS-Kernel · GitHub</a>
	OS 未使用
使用ボード	Renesas Envision KIT RPBRX65N（型名：RTK5RX65N2C00000BR） Renesas Envision Kit RPBRX72N（型名：RTK5RX72N0C00000BJ）

表 8.2 動作確認環境(Renesas Starter Kit)

項目	内容
統合開発環境	ルネサスエレクトロニクス製 e <sup>2</sup> studio 2021-10
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(CC-RX) V3.03.00 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 8.03.00.202004 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 4.20.1 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	リトルエンディアン
モジュールのバージョン	Ver.1.00
使用 OS	FreeRTOS <a href="#">Release Release RX MCUs FreeRTOS v1.0.1 comes from original 10.4.3 · renesas/FreeRTOS-Kernel · GitHub</a>
	OS 未使用
使用ボード	Renesas Starter Kit+ for RX65N-2MB（型名：RTK50565N2S10010BE） Renesas Starter Kit+ for RX72N（型名：RTK5572NNHS10000BE）

## 8.2 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「Could not open source file "platform.h"」エラーが発生します。

A: FIT モジュールがプロジェクトに正しく追加されていない可能性があります。プロジェクトへの追加方法をご確認ください。

- e<sup>2</sup> studio を使用している場合  
アプリケーションノート「RX ファミリ e<sup>2</sup> studio に組み込む方法 Firmware Integration Technology (R01AN1723)」

また、本 FIT モジュールを使用する場合、ボードサポートパッケージ FIT モジュール(BSP モジュール)もプロジェクトに追加する必要があります。BSP モジュールの追加方法は、アプリケーションノート「ボードサポートパッケージモジュール(R01AN1685)」を参照してください。

## 9. 参考ドキュメント

ユーザーズマニュアル：ソフトウェア

- ・ emWin Graphic Library with Graphical User Interface User Guide & Reference Manual  
(<https://www.segger.com/downloads/emwin/UM03001>)

ユーザーズマニュアル：ハードウェア

- ・ RXファミリ 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)  
(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)
- ・ RXファミリ 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。)

テクニカルアップデートの対応について

本モジュールにはテクニカルアップデートはありません。

## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2021.12.29	—	新規発行

## 製品ご使用上の注意事項

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

### 1. 静電気対策

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

### 2. 電源投入時の処置

電源投入時は、製品の状態は不定です。電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

### 3. 電源オフ時における入力信号

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

### 4. 未使用端子の処理

未使用端子は、「未使用端子の処理」に従って処理してください。CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。

### 5. クロックについて

リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

### 6. 入力端子の印加波形

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

### 7. リザーブアドレス（予約領域）のアクセス禁止

リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

### 8. 製品間の相違について

型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違っていると、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、ノイズ幅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、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、これに限りません。）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、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](http://www.renesas.com)

## 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。

## 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/)