

emWin for RX QEスタートアップ・設定ガイド



Using "QE for Display [RX]" RX65N Envision 評価ボード用

更新日:2024年6月

Constant Service Servic

Embedded Software / Hardware Solutions

試作から量産ツールまで

「emWin for RX」について

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

ロイヤリティー費用、量産制限等は有りません。









Copyright 2024 EmbiTeK Co., Ltd. All rights reserved.



事前準備

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

本セッションでは以下の開発環境を使用します。

- Host PC: Widows10 / Windows11
- 統合開発環境 e2 studio 2024-04
- ディスプレイ対応開発支援ツール パッケージ QE for Display[RX,RA] V3.2.0

EmbITeK

Copyright 2024 EmbiTeK Co., Ltd. All rights reserved.

以下のWebリンクから「統合開発環境e²studio 2024-04 インストーラ 」プログラムをダウンロードしてインストールします。

https://www.renesas.com/e2studio_download

統合開発環境 e² studio RXファミリ向け情報 設計・開発 調べる サポート ビデオ&トレーニング 概要 ダウンロード ドキュメント 詳細情報 ダウンロード Q 全種類 Start typing to filter results by title 分類 ♦ タイトル 🔶 日時 🔶 → アップデート – 統合開発 統合開発環境e² studio 2024-04 Windows用インストーラ 2024年4月22日 🔓 ログインしてダウンロード ZIP 1,587.34 MB 英語 環境 ★ アップデート – 統合開発 統合開発環境 e² studio 2024-04 Linux用 2024年4月22日 環境 ■ ログインしてダウンロード RUN 1.571.90 MB 英語 ↓ アップデート – 統合開発 統合開発環境e² studio 2024-04 macOS用インストーラ 2024年4月22日 ● ログインしてダウンロード XZ 700.09 MB 英語 環境

Copyright 2024 EmbiTeK Co., Ltd. All rights reserved.

事前準備:開発環境のインストール

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

🛃 Renesas e² studio 20	24-04 Setup	— 🗆 X
Renesas e² studio 2024	-04 Setup	RENESAS
インストールするデバイス・フ	ファミリを選択します。	
ようこそ	RA	
→ デバイス・ファミリ	Build Debug & Code Generation support for Renesas RA dev	vices
追加	Build Debug & Code Generation support for Renesas R7 dev	lices
ソノトウェア	Build Debug & Code Generation support for Renesas RI 78 d	evices
ライセンス	RX	
ショートカット	Build Debug & Code Generation support for Renesas RX dev	vices
概要	RH850	
インストール中	Debug support for Renesas RH850 devices	
結果	RE Build & Debug support for Renesas RF devices	
	RENESAS Svnerav	
	Build Debug & Code Generation support for Renesas Synerg	v devices
	ABRIAND DA	
	Build & Debug support for Renesas RISC-V MCU devices	
	Calact All	
	SCIECT AII	
<u>v202404151546</u>	User: All Users < Back Next >	インストール Cancel

民 Renesas e² studio 20	24-04 Setup		-	– 🗆 X
Renesas e² studio 2024	-04 Setup		RE	ΕΝΕΣΛΣ
インストールする追加ソフト	ウェアを選択します。		-	
ようこそ デバイス・ファミリ 迫加 ソフトウェア ライセンス ショートカット 概要 インストール中	 Renesas QE QE for AFE QE for Motor QE for Capacitive Touch QE for DEE[RA,RE,RA] QE for Display[RX,RA] QE for OTA Renesas AI Renesas Toolchains && Utilities Renesas CC-RX v3.06.00 	2.2.0 1.3.0 3.5.0 1.7.0 3.2.0 2.0.0 v3.06.00	23.6 MB	
結果	 Renesas E2 Self-Checking Program V1.02.00 Renesas E2 Lite Self-Checking Program V1.00.0 Renesas E1, E20 Self-Checking Program V1.01.0 GCC Toolchains && Utilities Renesas FSP Renesas FSP Renesas FSP v5.3.0 Renesas FSP v5.2.0 	v1.02.00 0 v1.00.00 00 v1.01.00 5.3.0 5.2.0	1000 B 1000 B 1000 B 121.8 MB 123.5 MB 879.5 MBΦ2	▼ ダウンロードが必要です。
<u>v202404151546</u>	User: All Users < Back	Next >	インストール	Cancel

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



Copyright 2024 EmbiTeK Co., Ltd. All rights reserved.

RX65N Envision Kit



マイコン:**RX65N MCU(RXv2コア 120 MHz動作)** 内蔵ROM:コードフラッシュメモリ:2MB 内蔵RAM:640KB デバッガI/F:E2 Liteエミュレータ

静電容量性タッチ機能付き4.3インチWQVGA TFTパネル

RX65N Envision Kitの詳細はこちら

<u>https://www.renesas.com/jp/ja/products/microcontrollers-microprocessors/rx-32-bit-performance-efficiency-mcus/rx65n-envision-kit-rx65n-envision-kit</u>



TASK-2

RX65N Envisionボード用 BSPソースコード作成

QE for Display [RX] のSmart Configuratorを使用して emWin+LCD表示に必要なソースコード及び プロジェクトを作成します。 実行手順: ・Envisionボード用プロジェクト作成 ・HEAPメモリ設定 ·emWin+ドライバモジュールの導入 ・ビデオバッファアドレスの設定 ・タッチパネルドライバ関連の設定 ・QE for Displayモジュールの設定 ・emWinを導入 ・LCDコントローラの導入 ・LCDインターフェース設定の確認 ・ファイル出力 emWinの初期設定、ファイル出力 ・メモリマップ設定 ・プロジェクトビルド確認 ・デバッグ設定



Copyright 2024 EmbiTeK Co., Ltd. All rights reserved.

Envisionボード用プロジェクト作成

_ノ プ(H)
is Debug
is RA 🔤 アウ
is RX
719-
S

e2StudioのSmart Configuratorを使用してemWin+ LCD表示に必要なソースコード及びプロジェクトを 作成します。

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





Copyright 2024 EmbiTeK Co., Ltd. All rights reserved.

プロジェクト作成:プロジェクト名、ボード名設定

①[プロジェクト名(P):]のテキストボックスにプロジェクト名 を入力し、②[次へ(N)>]をクリックします。





[Target Board:]の選択肢に「 EnvisionKitRX65N 」ボードを 選択してください。



Copyright 2024 EmbiTeK Co., Ltd. All rights reserved.

プロジェクト作成:ボード名、Configurator設定

① [Use Smart Configurator]のチェックボックスにチェックを入れます。 ②「次へ(N)>]をクリックします。





③ [Bare Metal - Minimal]プロジェクトテンプレートを選択します。 ④ [終了(F)]をクリックしてプロジェクトを作成します。



プロジェクト作成:スマート・コンフィグレータを開きます





"QE for Display [RX]"を使用します

6月

プロジェクト設定:スマート・コンフィグレータ設定(ボード情報)

「EnvisionKitRX65N」ボードの情報設定ファイルを最新版に設定します

i∰ rx65n_emwin_baremetal.scfg ×			
デバイス選択	じ		
デバイス選択	è 2		
ボード: EnvisionKitRX65N (V1.13) ~	e	— 🗆 X	
デバイス: R5F565NEDxFB ボード情報をダウンロードする	ボード定義ファイルのダウンロード ダウンロードするボード定義ファイルを選択してください。	2 J	🌼 rx65n_emwin_baremetal.scfg 🗡
3	タイトル ✓ Envision Kit for RX65N Board Description File	リビジョン ク すべて選択 1.14 選択をすべて解除	デバイス選択
		1.13	デバイス選択
概要 ボード クロック システム コンポーネント 端子 割り込み	モジュール・フォルダー・パス:		ボード· EnvisionKitRX65N (V/1 14)
	C:¥Users¥santo¥.eclipse¥com.renesas.platform_download¥	<i>∉</i> Boards	
			デバイス: R5F565NEDxFB
	5	ダウンロードキャンセル	<u>ボード情報をダウンロードする</u>

 ① [ボード] グループを選択します。
 ② ボード名[EnvisionKitRX65N]を確認します。
 ③ [ボード情報をダウンロードする] をクリックして ボード情報ファイルは最新版であることを確認します。 ④ ボード情報ファイルは古いバージョンの場合は最新版を選択して
⑤ [ダウンロード]をクリックして最新版を指定します。

EmbITeK

Copyright 2024 EmbiTeK Co., Ltd. All rights reserved.

プロジェクト設定:スマート・コンフィグレータ設定(DRW2Dエンジン用HEAPメモリ)





プロジェクト設定(コンポーネント):最新版のFITドライバとミドルウエアをダウンロード

🔯 *rx65n_emwin_baremetal.sc	🕑 コンポーネントの追加		- 🗆	×		
ソフトウェアコンポーネント設	ソフトウェアコンポーネントの選択 使用可能なコンポーネントの一覧から選択してくた	ざさい				
コンポ 辿 込 ほ ロ 田 当 1 セ セ フィルタ入力	カテゴリ 全て 機能 全て フィルタ		∠	~		
V 🗁 Startup	コンボーネント Sh	ort Name タ1 ー	「プ バージ ・ド生成 1.10」	·	e	
◆ 🗁 ジェネリック	DPCM compress/de-compress co r_s	s2_rx Fin	mware Inte 3.04	~	FITモジュールのダウンロード	- 1
💕 r_bsp	── 最新バージョンのみ表示				ダウンロードするFITモジュールを選択してください	
	☑ 重複する機能のコンポーネントを非表示				クイル ピセッシーへん リビジョン 発行日 -	すべて選択
	説明				✓ RXファミリ RX Driver Package Ver.1.42 R01AN716 Rev.1.42 2024-01-23 選邦	尺をすべて解除
	このソフトウェアコンポーネントは、8 ビットタイマ(TMR)の構成を提供しま	す。	^	统103.76.	
	外部イベントのカウントが可能なほか、2本のレジ	ズタとのコンペアマッチ信号	号により、カウンタのクリン	ד א	減り込み. 「 RX Driver Package のみ表示する	
2	最新版のFITドライバとミドルウェアをダウンロードす	3			モジュール・フォルダー・パス:	
					C:¥Users¥santo¥.eclipse¥com.renesas.platform_download¥FITModules	参照
					4 ダウンロード	キャンセル

① [コンポーネント追加] ボタンをクリックします。

②ダイアログから[最新版のFITドライバとミドルウエアをダウンロードする]リンクを実行して、

③、④ [RXファミリーRX Driver Package]の最新版をダウンロードします。

プロジェクト設定(コンポーネント):emWin+ドライバモジュールの導入

①コンポーネントの追加ダイアログの[機能]項目に「グラフィック ス」を選択して、

②以下のコンポーネントを選択します。

- Drawing 2D engine driver for RX (r_drw2d_rx)
- Graphic Library with Graphical User Interface (r_emwin_rx)
- Graphics LCD Controller Module (r_glcdc_rx)

③ [終了(F)]ボタンをクリックして選択したコンポーネントを プロジェクトに追加します。

e]	ンポ-	-ネントの追加			—		×
ソフト 使月	ウェ フ 用可能	アコンポーネントの選択 能なコンポーネントの一覧から選択して	てください			-	-
カテ	ゴリ	全て					~
機能	能	グラフィックス					\sim
フィ	ルタ						
יכ	ンポー	^ ネント	Short Name	タイプ	バージョン		
#	Drav	wing 2D engine driver for RX	r_drw2d_rx	Firmware Inte	1.12		
#	Grap	ohic Library with Graphical Use	r_emwin_rx	Firmware Inte	6.32.a.1.00		
+	Grap	ohic Library with One-Stop Dev	r_aeropoint_rx	Firmware Inte	1.00	2	
+	Grap	phics LCD Controller Module.	r_glcdc_rx	Firmware Inte	1.50		
	JPEC	G Decoder for Renesas MCUs.	r_jpegd_rx	Firmware Inte	2.06		11
+	JPEC	G Encoder for Renesas MCUs.	r_jpege_rx	Firmware Inte	1.01		
✓ ;	最新	バージョンのみ表示					
∠ <u>i</u>	重複	する機能のコンポーネントを非表示					
説明	抈						
依	存モ	ジュール: なし					^
Dr	rawir	ng 2D engine(DRW2D) driver for	RX MCUs				
							\sim
最新	新版(DFITドライバとミドルウェアをダウンロー	<u>ドする</u>				
<u>基</u> 2	本設定	<u>È</u>					
					3		
?)		< 戻る(B) 次^	(N) > 終	(F) キ	キャンセル	



Copyright 2024 EmbiTeK Co., Ltd. All rights reserved.

プロジェクト設定(コンポーネント):emWinモジュールの設定



①コンポーネント設定ウインドウから 「r_emwin_rx」を選択して、 以下の項目を編集します。

② Work area size for GUI = 81920

③ Start address of the Frame buffer 1 = 0x00800000 Frame buffer 2 = 0x00000100



プロジェクト設定(コンポーネント): I2Cドライバの設定(タッチパネルI/F用)

emWinでタッチパネルインターフェースに使用するIICチャネルを有効に設定します。 ※Envision RX65NボードのタッチパネルはIICチャネル「CH6」を使用しています。



①コンポーネント設定ウインドウから 「r_sci_iic_rx」を選択して、

② MCU supported channels for CH6 = Supported

③ [**コードの作成**] ボタンをクリックして 各コンポーネントのコードを作成します。





Copyright 2024 EmbiTeK Co., Ltd. All rights reserved.

プロジェクト設定:LCDインターフェース(QE for Display)

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

実行手順:

メニューから[Renesas Views]→[Renesas QE] →[LCD ワークフローRX,RA (QE)]コマンドを実行します。



EmbITeK

Copyright 2024 EmbiTeK Co., Ltd. All rights reserved.

「LCDワークフローRX,RA(QE)」ワークフローの設定を行います。

① [プロジェクトの選択]をクリックして

対象のプロジェク「QEDisp_RX65N_Envision」
 を選択します。

③ [GUI 描画ツールの選択]をクリックして

④ GUI 作成ツールに [emWin を使用する] を選択し ます。



🔯 rx65n_emwin_baremetal.scfg 🗣 *LCDワ−クフロ− RX,RA	(QE) ×	
🗎 🕽 💻 QE for Display		
⊘ 1. 準備	2. LCDの調整	3. 画面表示の作成
1. 準備 🗸 👻	GUI描	画ツールの選択
 ✓ プロジェクトの選択 ✓ LCDの選択 	LCDに表示するGUIを作成する	ソールを選択してください
 ✓ GUI描画ツールの選択 	emWir	を使用する
2. LCDの調整 -		



Copyright 2024 EmbiTeK Co., Ltd. All rights reserved.

プロジェクト設定:LCDインターフェース(QE for Display)



① [LCDコントローラの導入]をクリックして

②「導入済み」であることを確認します。

EmbITeK

Copyright 2024 EmbiTeK Co., Ltd. All rights reserved.

プロジェクト設定:LCDインターフェース(QE for Display)



① [LCDの表示調整]をクリックしてLCDの表示調整設定ダイアログを開きます。

②「TCON/LCD設定」ボタンをクリックして「ディスプレイ調整RX(QE)」ダイアログを開きます。

プロジェクト設定:LCDインターフェース設定内容を確認します









Copyright 2024 EmbiTeK Co., Ltd. All rights reserved.

プロジェクト設定: LCDインターフェース (QE for Display)

「LCDワークフローRX,RA(QE)」ワークフローのLCD設定を完了します。



Copyright 2024 EmbiTeK Co., Ltd. All rights reserved.

プロジェクト設定:GUI描画ツールの設定(emWinの初期設定)

🔅 rx65n emwin baremetal.scfg

🕒 🔻 🖇 🗖 E

QE for Display \odot 1. 準備 3. 画面表示の作成 \odot 2. LCDの調整 1. 準備 $\mathbf{\wedge}$ • GUI描画ツールの設定 $\mathbf{\vee}$ プロジェクトの選択 (~) GUI描画ツールを使用するための設定を行います LCDの選択 **~**) GUI描画ツールの選択 (~) GUI描画ツールの設定: 2. LCDの調整 2 情報設定 LCDコントローラの導入 **~**) LCDの表示調整 **~**) ファイル出力 3. 画面表示の作成 • フォルダ指定 \square GUI描画ツールの導入 \checkmark GUI描画ツールの設定 **GUIの作成** GUI描画ツールの設定をする 実装 emWinお上びAeronoint CUIの設定を行います

① [GUI描画ツールの設定]をクリックしてダイアログから

「情報設定」ボタンをクリックして「emWinの設定」ダイアログを開きます。 (2)



Copyright 2024 EmbiTeK Co., Ltd. All rights reserved.

プロジェクト設定:emWinの設定(初期設定)

國 emWinの設定

甘木信起		
季本時報 使用ボード	RX65N_ENVISION_KIT	
画面サイズ(横)	480	
画面サイズ(縦)	272	
画面の基本設定 LCDの向き ピクセル当たりのカラー深度	ORIENTATION_0 ~ 16bits ~	
端子設定 ✓ LCDパネルのリセット端子	ポート:6 ビット:3	
✓ LCDパネルのバックライト端子	ポート: 6 ビット: 6	
✓LCDパネル搭載のタッチICのリセット端子	ポート: 0 〜 ビット: 7 〜	
メモリ設定 フレームバッファ1アドレス フレームバッファ2アドレス GUIで使用する最大メモリサイズ	0x00800000 0x00000100 81920	1 DRW2D
<u> </u>		
タッチ機能の使用	使用する ~	
タッチ接続インタフェース	SCI_IIC ×	
タッチインタフェースのチャネル番号	6	2D描画エンジンDRW2Dの使用有無を設定します。DRW2Dを使用する
タッチパネルのスレーブアドレス	0x38	ことで高速な描画が可能になります。
マルチタッチ機能の使用	使用しない ~	DRW2Dを使用する場合は、ビクセルのだりのカラー深度をTobitsまたは 32bitsに設定してください。
タッチパネルのポイントの最大数	10	
DRW2D機能 DRW2Dの使用	使用する 、	2
		OK キャンセル ヘルプ(H)

 ①[フレームバッファ2アドレス]を「Ox00000100」に 設定します。
 ②「OK」ボタンをクリックして設定を保存します。

ビデオフレームバッファ1アドレス: 0x00800000 ビデオフレームバッファ2アドレス: 0x00000100 emWin用メモリプールサイズ: 81920 Bytes タッチ機能の使用:使用する マルチタッチ機能:使用しない (シングルタッチモード) DRW2D機能:使用する



Copyright 2024 EmbiTeK Co., Ltd. All rights reserved.

プロジェクト設定:GUI描画ツールの設定(emWin設定)

rx65n emwin baremetal.scfg

🕩 🔻 🖇 🖵 E

QE for Display \odot 3. 画面表示の作成 1. 準備 2. LCDの調整 \odot 1. 準備 -GUI描画ツールの設定 \checkmark プロジェクトの選択 E 🕏 7 **(~**) Pairie 100 - 200 - GUI描画ツールを使用するための設定を行います ✓ [™] rx65n emwin baremetal [HardwareDebug] LCDの選択 > 🖑 バイナリー GUI描画ツールの選択 ✓) GUI描画ツールの設定: Includes ✓ ²⁹/₂ src 2. LCDの調整 情報設定 > 🗁 smc_aen LCDコントローラの導入 Image_emwin_config.h Image_config.h LCDの表示調整 In the second ファイル出力 2 3. 画面表示の作成 Ising the second sec -フォルダ指定 \square GUI描画ツールの導入 💷 コンソール 🗡 GUI描画ツールの設定 QE for Display **GUIの作成** ファイルを生成しました。 GUI描画ツールの設定をする (1) C:\Workspace\RX-emWin\RX65N_Envision_Kit\rx65n_emwin_baremetal\src\r_lcd_timing.h (2) C:\Workspace\RX-emWin\RX65N_Envision_Kit\rx65n_emwin_baremetal\src\r_image_config.h 実装 ファイルを生成しました。 emWinおよびAeronoint GUIの設定 (1) C:\Workspace\RX-emWin\RX65N_Envision_Kit\rx65n_emwin_baremetal\src\qe_emwin_config.h

① [GUI描画ツールの設定]をクリックしてダイアログから

「ファイル出力」ボタンをクリックしてQEコンフィグレーション設定ファイル「qe_emwin_config.h」を作成します。

EmbITeK

Copyright 2024 EmbiTeK Co., Ltd. All rights reserved.

プロジェクト設定:リンカーマップのビルド設定



el RX - e ² studio	「プロジェクト→プロジェクトのビルド」メニューコマンドでプロジェクト
ファイル(F) 編集(E) ナビゲート(N) 検索(A) プロジェクト(P) Renesas \ 🔚 🛞 ▼ 🔦 ▼ 🥔 🔄 🗐 🔅 ▼ 🂁 ▼ 🎦 プロジェクト・エクスプローラー ×	をビルドして結果を確認します。
 includes includes is nc_gen includes, includes is mc_gen include, 	<pre>CDT Y/J-W × CDT Y/J-W [rx65n_emwin_baremetal] Scanning and building file:/src/smc_gen/r_drw2d_rx/src/rx/dave_base_rx.c Scanning and building file:/src/smc_gen/r_drw2d_rx/src/rx/dave_memory_rx.c Scanning and building file:/src/smc_gen/r_bsy/mcu/rx65n/vecttbl.c Scanning and building file:/src/smc_gen/r_bsy/mcu/rx65n/vecttbl.c Scanning and building file:/src/smc_gen/r_bsy/mcu/rx65n/vecttbl.c Scanning and building file:/src/smc_gen/r_bsy/mcu/rx65n/hsetup.c Scanning and building file:/src/smc_gen/r_bsy/mcu/rx65n/mcu_interrupts.c Scanning and building file:/src/smc_gen/r_size_iic_rx65n/mcu_interrupts.c Scanning and building file:/src/smc_gen/r_size_iic_rx65n/mcu_interrupts.c Scanning and building file:/src/smc_gen/r_size_iic_rx/src/targets/rx65n/r_sci_iic_rx65n.c Invoking Linker: rx65n_emwin_baremetal.abs Renesas Optimizing Linker Completed Loading input file rx65n_emwin_baremetal.abs Parsing the ELF input file rx65n_emwin_baremetal.x Invoking Converting the output ELF image Saving the DWARF information Constructing the output file rx65n_emwin_baremetal.x Invoking Converter: rx65n_emwin_baremetal.mot Renesas Optimizing Linker Completed Build complete.] 16:54:55 Build Finished. 0 errors, 0 warnings. (took 5s.919ms)</pre>



プロジェクト設定:デバッグ設定

「実行→デバッグの構成」メニューコマンドでデバッグ構成ダイアログを開きます。

プロジェクトのGDB Hardware Debugging構成を選択ます。

🗃 デバッグ構成			- 🗆 X
構成の作成、管理、および実行			Ť
 ご ご い こ こ こ こ こ こ こ こ こ こ こ こ こ こ こ こ こ	名前(N): rx65n_emwin_baremetal HardwareDeb マロジェクト(P): rx65n_emwin_baremetal	ug ソース	参照(B)
 GDB Simulator Debugging (RH850) GDB ハードウェア・デバッギング Renesas GDB Hardware Debugging アx65n_emwin_baremetal HardwareDebug 	C/C++ アプリケーション: HardwareDebug/rx65n_emwin_baremetal.x	変数(V) プロジェクトの検索(H)	参照(R)
<mark>■ Renesas Simulator Debugging (RX, RL78)</mark> ■ 起動グループ	起動前に必要に応じてビルド <u>Build Configuration</u> : Select Automatically ○ 自動ビルドを有効にする ④ ワークスペース設定の使用	○ 自動ビルドを無効にする <u>ワークスペース設定の構成</u>	~

EmbITeK

Copyright 2024 EmbiTeK Co., Ltd. All rights reserved.

プロジェクト設定:デバッグ設定



Debug hardware: E2 Lite (RX)	R5F565NE	
GDB Settings Connection Settings デバッグ・ツール設定	(Auto)	
	JTag	~
JTag クロック周波数[MHz]	6.00	~
Fine ボーレート[Mbps]	1.50	\sim
ホット・プラグ	いいえ	~
◇ 電源		
エミュレーターから電源を供給する (MAX 200mA)	いいえ	~
供給電圧 (V)	3.3	\sim
Y CPU 動作モード		
レジスター設定	シングルチップ	\checkmark
モード端子	シングルチップ・モード	Ň
起動バンクを変更する	いいえ	4
	前回保管した状態に戻す(\/)	適田(¥)

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

メイン・クロック・ソース	EXTAL
EXTAL 周波数[MHz]	12.0000
接続タイプ	JTAG
エミュレーターから電源を供給する	いいえ

④ [適用] ボタンをクリックして設定を保存します。



bug

Copyright 2024 EmbiTeK Co., Ltd. All rights reserved.

画面表示の作成 (事前設定)

emWinアプリケーションのスタートアップコードを設定します。



Copyright 2024 EmbiTeK Co., Ltd. All rights reserved.



画面表示の作成:emWin UI 作成方法

オプション1

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

🗮 AppWard V0 50b, 6.10 (Reta) - Temp, cut, 3								
Elle Edit Project Resource	Help							
Add objects		Editor					Properties	
		Q -		Q 1:1			Id ID_BUTTON_BR_FAN_ON	
	\square			Temperature	Cookel		Position: 284 234 size: 50 26	
Window	Box		Deemi	Fact Torra i	Control	SECCER		
	0		RUUIII.	Fail. Terrip	- <u>1</u> -1	(W SEGGEN)		
	\sim		Bedroom	Off 0 °C	Temperal	ture	← → ← →	
Button	Image							
			Living Room	Off 0 °C				
Abc					0°C			
Test	Switch		Kitchen	Un U'C				
			Eamily Doom	0.65 0.20				
			Failing Room		Fan			
			Dining Room	Off 0 °C	O O Off	Auto	Top: Width:	
Screen	Sider				0=0=0=		-1 C Bottom: 0 C Height	
		Interaction	s				0 🗘 26 🛟	
		Æ					- Sattant colore	
Rotary				Signal	lab	Pacoline	Unressed Pressed Disabled	
Hierarchic tree		× / 1	LEUTTON_BR	WM_NOTIFICATION_CLICKED	APPW_JOB_SETVIS	ID_BOX_DR_BUTTON_HL	× 2 × 2 333339 2	
			BUTTON_BR	WM_NOTIFICATION_CLICKED	APPW_JOB_SETVIS	ID_WINDOW_BR_CTRLS	Set background colors	
Object Id		× / II	_BUTTON_BR	WM_NOTIFICATION_CLICKED	APPW_JOB_SETVIS	ID_WINDOW_KT_CTRLS	▼ Set bitmaps X	
Window ID_Wit	NDOW_BR_CTRLS - 2061	× / II	DEUTTON_BR	WM_NOTIFICATION_CLICKED	APPW_JOB_SETVIS APPW_JOB_SETVIS	ID_WINDOW_FR_CTRLS	Unpressed Pressed Disabled	
Rotary ID_R0 Text ID_TE	TARY_BR - 2051 KT_TEMP_BR - 2052	X	BUTTON_BR_FAN_ON	WM_NOTIFICATION_CLICKED	APPW_JOB_SET	ID_BUTTON_BR_FAN_ON	^*	
Button ID_BUT	TTON_BR_FAN_ON - 2067	× ii	BUTTON_BR_FAN_ON	WM_NOTIFICATION_CLICKED	APPW_JOB_CLEAR	ID_BUTTON_BR_FAN_AUTO		
Button ID_BU	TTON_BR_FAN_AUTO - 2068	× II	BUTTON_BR_FAN_ON	WM_NOTIFICATION_CLICKED WM_NOTIFICATION_CLICKED	APPW_JO8_SETTEXT APPW_JO8_SET	ID_TEXT_BR_FAN ID_BUTTON_BR_FAN_OFF		
► Window ID_Wit	NDOW_KT_CTRLS - 2081	× I	BUTTON BR FAN OFF	WM_NOTIFICATION_CLICKED	APPW_JOB_CLEAR	ID_BUTTON_BR_FAN_ON	Set bitmap alignment X	
Text	Fonts	× / ii	_BUTTON_BR_FAN_OFF	WM_NOTIFICATION_CLICKED	APPW_JOB_SETTEXT	ID_TEXT_BR_FAN	Set text	
₽	Tt	× u	BUTTON_BR_FAN_AUTO	WM_NOTIFICATION_CLICKED WM_NOTIFICATION_CLICKED	APPW_JOB_SET APPW_JOB_CLEAR	ID_BUTTON_BR_FAN_AUTO ID_BUTTON_BR_FAN_ON	Set text alignment X	
Images	Variables	×	BUTTON BR FAN AUTO	WM_NOTIFICATION_CLICKED	APPW_JOB_CLEAR	ID_BUTTON_BR_FAN_OFF	▼ Set font X	
	(*)	× / 1	BUTTON_LR	WM_NOTIFICATION_CLICKED	APPW_JOB_SETVIS	ID_BOX_LR_BUTTON_HL	Roboto_16_Normal_EXT_AA4.xbf	
	(A)	X / II	BUTTONLIR	WM_NOTIFICATION_CLICKED	APPW_JOB_SETVIS	ID_BOX_BR_BUT TON_HL	the second	

※ AppWizardツールのインストールが必要です。

オプション2

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



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



Copyright 2024 EmbiTeK Co., Ltd. All rights reserved.

オプション1 AppWizardから画面作成:事前設定

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



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



2024年

6月

P.34



Copyright 2024 EmbiTeK Co., Ltd. All rights reserved.

オプション1 AppWizardから画面作成:emWinスタートアップコード

```
AppWizardツールの「File→ Export&Save」コマンドでUI
プロジェクトを作成すると、「aw¥Source」フォルダ下に
GUI初期化ハンドラソースファイル(APPW_MainTask.c)
も作成されます。
```

```
エントリハンドラ関数:MainTask()
```



<pre>File : APPW_MainTask.c↓ Purpose : APPWIZARD application entry ppint↓ File : APPWIZARD application entry ppint↓ File : # MainTask.c↓ File : # MainTask.c</pre>	11,	
<pre>11 Purpose : Arrw</pre>	12	File · APPW MainTask ch
<pre>Purpose : APPWIZARD application entry ppints */↓ 4 </pre>	12	
<pre>14END-OF-HEADER</pre>	13	Purpose : APPWIZARD application entry p_{pint}
<pre>15 */↓ 16 ↓ 17 #include "Generated/Resource.h"↓ 18 ↓ 19 /************************************</pre>	14	END_OF_HEADER
<pre>16 ↓ 17 #include "Generated/Resource.h"↓ 18 ↓ 19 /************************************</pre>	15	*/↓
<pre>17 #include "Generated/Resource.h"↓ 18 ↓ 19 /************************************</pre>	16	\downarrow
<pre>11 * Interdue Generated/Resource.In * 12 14 14 15 15 16 17 17 17 17 17 17 17 17 17 17 17 17 17</pre>	17	tipelude "Constant (Decourse b"
<pre>18 ↓ 19 /************************************</pre>	11	# Include Generated/Resource. In the
<pre>19 /************************************</pre>	18	\downarrow
<pre>20 *↓ 21 * MainTask↓ 22 */* 23 void MainTask(void) {↓ 24 //↓ 25 // Setup configuration dependent pointers↓ 26 //↓ 27 APPW_X_Setup();↓ 28 //↓ 29 // Initialize AppWizard↓ 30 //↓ 31 APPW_Init(APPW_PROJECT_PATH);↓ 32 //↓ 33 // Create all persistent screens except initial screen↓ 34 //↓ 35 APPW_CreatePersistentScreens();↓ 36 //↓ 37 // Create initial screen↓ 38 //↓ 39 APPW_CreateRoot(APPW_INITIAL_SCREEN, WM_HBKWIN);↓ 40 //↓ 41 //and keep it alive↓ 42 //↓ 43 while (1) {↓ 44 while (GUI_Exec1()) {↓ 45 APPW_Exec();↓ 46 }↓↓ 47 APPW_Exec();↓ 48 GUI_X_Delay(5);↓ 49 }↓ 50 }↓</pre>	19	/**************************************
<pre>21 * MainTask↓ 22 */* 23 void MainTask(void) {↓ 24 //↓ 25 // Setup configuration dependent pointers↓ 26 //↓ 27 APPW_X_Setup();↓ 28 //↓ 29 // Initialize AppWizard↓ 30 //↓ 31 APPW_Init(APPW_PROJECT_PATH);↓ 32 //↓ 33 // Create all persistent screens except initial screen↓ 34 //↓ 35 APPW_CreatePersistentScreens();↓ 36 //↓ 37 // Create initial screen↓ 38 //↓ 39 APPW_CreateRoot(APPW_INITIAL_SCREEN, WM_HBKWIN);↓ 40 //↓ 41 //and keep it alive↓ 42 //↓ 43 while (1) {↓ 44 while (GUI_Exec1()) {↓ 45 APPW_Exec();↓ 46 }↓ 47 APPW_Exec();↓ 48 GUI_X_Delay(5);↓ 49 }↓ 50 }↓</pre>	20	*
<pre>22 */* 23 void MainTask(void) {↓ 24 //↓ 25 // Setup configuration dependent pointers↓ 26 //↓ 27 APPW_X_Setup();↓ 28 //↓ 29 // Initialize AppWizard↓ 30 //↓ 31 APPW_Init(APPW_PROJECT_PATH);↓ 32 //↓ 33 // Create all persistent screens except initial screen↓ 34 //↓ 35 APPW_CreatePersistentScreens();↓ 36 //↓ 37 // Create initial screen↓ 37 // Create initial screen↓ 38 //↓ 39 APPW_CreateRoot(APPW_INITIAL_SCREEN, WM_HBKWIN);↓ 40 //↓ 41 //and keep it alive↓ 42 //↓ 43 while (1) {↓ 44 while (GUI_Exec1()) {↓ 45 APPW_Exec();↓ 46 }↓ 47 APPW_Exec();↓ 48 GUI_X_Delay(5);↓ 49 }↓ 50 }↓ </pre>	21	* MainTask↓
<pre>void MainTask(void) {↓ void MainTask(void MainTask(vo</pre>		8/2
<pre>24 //↓ 25 //↓ 26 //↓ 27 APPW_X_Setup();↓ 28 //↓ 29 // Initialize AppWizard↓ 30 //↓ 31 APPW_Init(APPW_PROJECT_PATH);↓ 32 //↓ 33 // Create all persistent screens except initial screen↓ 34 //↓ 35 APPW_CreatePersistentScreens();↓ 36 //↓ 37 // Create initial screen↓ 38 //↓ 39 APPW_CreateRoot(APPW_INITIAL_SCREEN, WM_HBKWIN);↓ 40 //↓ 41 //and keep it alive↓ 42 //↓ 43 while (1) {↓ 44 while (1) {↓ 45 APPW_Exec();↓ 46 }↓ 47 APPW_Exec();↓ 48 GUI_X_Delay(5);↓ 49 }↓ 50 }↓</pre>	22	void MainTack(void) f
<pre>24 //↓ 25 // Setup configuration dependent pointers↓ 26 //↓ 27 APPW_X_Setup();↓ 28 //↓ 29 // Initialize AppWizard↓ 30 //↓ 31 APPW_Init(APPW_PROJECT_PATH);↓ 32 //↓ 33 //↓ Create all persistent screens except initial screen↓ 34 //↓ 35 APPW_CreatePersistentScreens();↓ 36 //↓ 37 // Create initial screen↓ 38 //↓ 39 APPW_CreateRoot(APPW_INITIAL_SCREEN, WM_HBKWIN);↓ 40 //↓ 41 //and keep it alive↓ 42 //↓ 43 while (1) {↓ 44 while (GUI_Execl()) {↓ 45 APPW_Exec();↓ 46 }↓ 47 APPW_Exec();↓ 48 GUI_X_Delay(5);↓ 49 }↓ 50 }↓</pre>	23	
<pre>25 // Setup configuration dependent pointers↓ 26 //↓ 27 APPW_X_Setup();↓ 28 //↓ 30 //↓ 31 APPW_Init(APPW_PROJECT_PATH);↓ 32 //↓ 33 // Create all persistent screens except initial screen↓ 34 //↓ 35 APPW_CreatePersistentScreens();↓ 36 //↓ 37 // Create initial screen↓ 38 //↓ 39 APPW_CreateRoot(APPW_INITIAL_SCREEN, WM_HBKWIN);↓ 40 //↓ 41 //and keep it alive↓ 42 //↓ 43 while (1) {↓ 44 while (GUI_Exec1()) {↓ 45 APPW_Exec();↓ 46 }↓ 47 APPW_Exec();↓ 48 GUI_X_Delay(5);↓ 49 }↓ 50 }↓</pre>	24	//↓
<pre>26 //↓ 27 APPW_X_Setup();↓ 28 //↓ 29 // Initialize AppWizard↓ 30 //↓ 31 APPW_Init(APPW_PROJECT_PATH);↓ 32 //↓ 33 //↓ 34 //↓ 35 APPW_Create all persistent screens except initial screen↓ 36 //↓ 37 // Create initial screen↓ 38 //↓ 39 APPW_CreateRoot(APPW_INITIAL_SCREEN, WM_HBKWIN);↓ 40 //↓ 41 //and keep it alive↓ 42 //↓ 43 while (1) {↓ 44 while (GUI_Exec1()) {↓ 45 APPW_Exec();↓ 46 }↓ 47 APPW_Exec();↓ 48 GUI_X_Delay(5);↓ 49 }↓ 50 }↓</pre>	25	// Setup configuration dependent pointers $ ightarrow$
<pre>APPW_X_Setup();↓ //↓ // Initialize AppWizard↓ //↓ APPW_Init(APPW_PROJECT_PATH);↓ //↓ //↓ APPW_Create all persistent screens except initial screen↓ //↓ APPW_CreatePersistentScreens();↓ //↓ APPW_CreateRoot(APPW_INITIAL_SCREEN, WM_HBKWIN);↓ //↓ APPW_CreateRoot(APPW_INITIAL_SCREEN, WM_HBKWIN);↓ //↓ APPW_CreateRoot(APPW_INITIAL_SCREEN, WM_HBKWIN);↓ //↓ Ample (1) {↓ while (1) {↓ while (GUI_Exec1()) {↓ APPW_Exec();↓ GUI_X_Delay(5);↓ } }↓ </pre>	26	//↓
<pre>28 //↓ 29 // Initialize AppWizard↓ 30 //↓ 31 APPW_Init(APPW_PROJECT_PATH);↓ 32 //↓ 33 // Create all persistent screens except initial screen↓ 34 //↓ 35 APPW_CreatePersistentScreens();↓ 36 //↓ 37 // Create initial screen↓ 38 //↓ 39 APPW_CreateRoot(APPW_INITIAL_SCREEN, WM_HBKWIN);↓ 40 //↓ 41 //and keep it alive↓ 42 //↓ 43 while (1) {↓ 44 while (GUI_Exec1()) {↓ 45 APPW_Exec();↓ 46 }↓↓ 47 APPW_Exec();↓ 48 GUI_X_Delay(5);↓ 49 }↓ 50 }↓ </pre>	27	APPW X Setup():
<pre>// Initialize AppWizard↓ //↓ APPW_Init(APPW_PROJECT_PATH);↓ //↓ //↓ APPW_Create all persistent screens except initial screen↓ //↓ APPW_CreatePersistentScreens();↓ //↓ APPW_Create initial screen↓ //↓ APPW_CreateRoot(APPW_INITIAL_SCREEN, WM_HBKWIN);↓ //↓ APPW_CreateRoot(APPW_INITIAL_SCREEN, WM_HBKWIN);↓ //↓ while (1) {↓ while (1) {↓ while (1) {↓ APPW_Exec();↓ APPW_Exec();↓ GUI_X_Delay(5);↓ } </pre>	28	///
<pre>29 // Initiatize AppWizard↓ 30 //↓ 31 APPW_Init(APPW_PROJECT_PATH);↓ 32 //↓ 33 // Create all persistent screens except initial screen↓ 34 //↓ 35 APPW_CreatePersistentScreens();↓ 36 //↓ 37 // Create initial screen↓ 38 //↓ 39 APPW_CreateRoot(APPW_INITIAL_SCREEN, WM_HBKWIN);↓ 40 //↓ 41 //and keep it alive↓ 42 //↓ 43 while (1) {↓ 44 while (GUI_Execl()) {↓ 45 APPW_Exec();↓ 46 }↓ 47 APPW_Exec();↓ 48 GUI_X_Delay(5);↓ 49 }↓ 50 }↓ </pre>	20	//*
<pre>30 //↓ 31 APPW_Init(APPW_PROJECT_PATH);↓ 32 //↓ 33 // Create all persistent screens except initial screen↓ 34 //↓ 35 APPW_CreatePersistentScreens();↓ 36 //↓ 37 // Create initial screen↓ 38 //↓ 39 APPW_CreateRoot(APPW_INITIAL_SCREEN, WM_HBKWIN);↓ 40 //↓ 41 //and keep it alive↓ 42 //↓ 43 while (1) {↓ 44 while (GUI_Exec1()) {↓ 45 APPW_Exec();↓ 46 }↓ 47 APPW_Exec();↓ 48 GUI_X_Delay(5);↓ 49 }↓ 50 }↓</pre>	29	// Initiatize Appwizard
<pre>31 APPW_Init(APPW_PROJECT_PATH);↓ 32 //↓ 33 // Create all persistent screens except initial screen↓ 34 //↓ 35 APPW_CreatePersistentScreens();↓ 36 //↓ 37 // Create initial screen↓ 38 //↓ 39 APPW_CreateRoot(APPW_INITIAL_SCREEN, WM_HBKWIN);↓ 40 //↓ 41 //and keep it alive↓ 42 //↓ 43 while (1) {↓ 44 while (GUI_Exec1()) {↓ 45 APPW_Exec();↓ 46 }↓ 47 APPW_Exec();↓ 48 GUI_X_Delay(5);↓ 49 }↓ 50 }↓</pre>	30	
<pre>32 //↓ 33 // Create all persistent screens except initial screen↓ 34 //↓ 35 APPW_CreatePersistentScreens();↓ 36 //↓ 37 // Create initial screen↓ 38 //↓ 39 APPW_CreateRoot(APPW_INITIAL_SCREEN, WM_HBKWIN);↓ 40 //↓ 41 //and keep it alive↓ 42 //↓ 43 while (1) {↓ 44 while (GUI_Exec1()) {↓ 45 APPW_Exec();↓ 46 }↓ 47 APPW_Exec();↓ 48 GUI_X_Delay(5);↓ 49 }↓ 50 }↓</pre>	31	APPW_Init(APPW_PROJECT_PATH);↓
<pre>33 // Create all persistent screens except initial screen↓ 34 //↓ 35 APPW_CreatePersistentScreens();↓ 36 //↓ 37 // Create initial screen↓ 38 //↓ 39 APPW_CreateRoot(APPW_INITIAL_SCREEN, WM_HBKWIN);↓ 40 //↓ 41 //and keep it alive↓ 42 //↓ 43 while (1) {↓ 44 while (GUI_Exec1()) {↓ 45 APPW_Exec();↓ 46 }↓ 47 APPW_Exec();↓ 48 GUI_X_Delay(5);↓ 49 }↓ 50 }↓ </pre>	32	//↓
<pre>34 //↓ 35 APPW_CreatePersistentScreens();↓ 36 //↓ 37 // Create initial screen↓ 38 //↓ 39 APPW_CreateRoot(APPW_INITIAL_SCREEN, WM_HBKWIN);↓ 40 //↓ 41 //and keep it alive↓ 42 //↓ 43 while (1) {↓ 44 while (GUI_Exec1()) {↓ 45 APPW_Exec();↓ 46 }↓ 47 APPW_Exec();↓ 48 GUI_X_Delay(5);↓ 49 }↓ 50 }↓</pre>	33	// Create all persistent screens except initial screen \downarrow
	24	
APPW_CreatePersistentScreens(); $//\downarrow$ $//\downarrow$ $//\downarrow$ APPW_CreateRoot(APPW_INITIAL_SCREEN, WM_HBKWIN); $//\downarrow$ $//\downarrow$ $//\downarrow$ $//\downarrow$ $//\downarrow$ $//\downarrow$ while (1) { \downarrow $//\downarrow$	34	
<pre>36 //↓ 37 // Create initial screen↓ 38 //↓ 39 APPW_CreateRoot(APPW_INITIAL_SCREEN, WM_HBKWIN);↓ 40 //↓ 41 //and keep it alive↓ 42 //↓ 43 while (1) {↓ 44 while (GUI_Exec1()) {↓ 45 APPW_Exec();↓ 46 }↓ 47 APPW_Exec();↓ 48 GUI_X_Delay(5);↓ 49 }↓ 50 }↓</pre>	35	APPw_CreatePersistentScreens();↓
<pre>37 // Create initial screen↓ 38 //↓ 39 APPW_CreateRoot(APPW_INITIAL_SCREEN, WM_HBKWIN);↓ 40 //↓ 41 //and keep it alive↓ 42 //↓ 43 while (1) {↓ 43 while (1) {↓ 44 while (GUI_Exec1()) {↓ 45 APPW_Exec();↓ 46 }↓ 47 APPW_Exec();↓ 48 GUI_X_Delay(5);↓ 49 }↓ 50 }↓</pre>	36	
<pre>38 //↓ 39 APPW_CreateRoot(APPW_INITIAL_SCREEN, WM_HBKWIN);↓ 40 //↓ 41 //and keep it alive↓ 42 //↓ 43 while (1) {↓ 44 while (GUI_Exec1()) {↓ 45 APPW_Exec();↓ 46 }↓ 47 APPW_Exec();↓ 48 GUI_X_Delay(5);↓ 49 }↓ 50 }↓</pre>	37	// Create initial screen \downarrow
<pre>39 APPW_CreateRoot(APPW_INITIAL_SCREEN, WM_HBKWIN);↓ 40 //↓ 41 //and keep it alive↓ 42 //↓ 43 while (1) {↓ 43 while (GUI_Exec1()) {↓ 44 while (GUI_Exec1()) {↓ 45 APPW_Exec();↓ 46 }↓ 47 APPW_Exec();↓ 48 GUI_X_Delay(5);↓ 49 }↓ 50 }↓</pre>	38	//↓
<pre>40 //↓ 41 //and keep it alive↓ 42 //↓ 43 while (1) {↓ 44 while (GUI_Exec1()) {↓ 45 APPW_Exec();↓ 46 }↓ 47 APPW_Exec();↓ 48 GUI_X_Delay(5);↓ 49 }↓ 50 }↓</pre>	39	APPW CreateRoot (APPW INITIAL SCREEN WM HRKWIN).
<pre>40 //↓ 41 //and keep it alive↓ 42 //↓ 43 while (1) {↓ 44 while (GUI_Exec1()) {↓ 45 APPW_Exec();↓ 46 }↓ 47 APPW_Exec();↓ 48 GUI_X_Delay(5);↓ 49 }↓ 50 }↓</pre>	40	
<pre>41 //and keep it alive↓ 42 //↓ 43 while (1) {↓ 44 while (GUI_Exec1()) {↓ 45 APPW_Exec();↓ 46 }↓ 47 APPW_Exec();↓ 48 GUI_X_Delay(5);↓ 49 }↓ 50 }↓</pre>	40	
<pre>42 //↓ 43 while (1) {↓ 44 while (GUI_Exec1()) {↓ 45 APPW_Exec();↓ 46 }↓ 47 APPW_Exec();↓ 48 GUI_X_Delay(5);↓ 49 }↓ 50 }↓</pre>	41	//and keep it alive↓
<pre>43 while (1) {↓ 44 while (GUI_Exec1()) {↓ 45 APPW_Exec();↓ 46 }↓ 47 APPW_Exec();↓ 48 GUI_X_Delay(5);↓ 49 }↓ 50 }↓ </pre>	42	
<pre>44 while (GUI_Exec1()) {↓ 45 APPW_Exec();↓ 46 }↓ 47 APPW_Exec();↓ 48 GUI_X_Delay(5);↓ 49 }↓ 50 }↓</pre>	43	while (1) { \downarrow
<pre>45 APPW_Exec();↓ 46 }↓ 47 APPW_Exec();↓ 48 GUI_X_Delay(5);↓ 49 }↓ 50 }↓</pre>	44	while (GUI Exec1()) {↓
<pre>46</pre>	45	
46 }↓ 47 APPW_Exec();↓ 48 GUI_X_Delay(5);↓ 49 }↓ 50 }↓	10	
47 APPW_Exec();↓ 48 GUI_X_Delay(5);↓ 49 }↓ 50 }↓	46	
48 GUI_X_Delay(5);↓ 49 }↓ 50 }↓	47	APPW_Exec();↓
49 }↓ 50 }↓	48	GUI_X_Delay(5);↓
50 }↓	49	$\}\downarrow$
	50	}↓
	-1	



Copyright 2024 EmbiTeK Co., Ltd. All rights reserved.

EmbiTeK Co., Ltd. All rights reserved.

アプリケーションプログラムのmain()関数からGUIハンドラ「MainTask()関数」を実行します。

```
emWinスタートアップ実装事例:
                                                         2
                                                          *↓
                                                                         : rx65n_emwin_baremetal.c↓
                                                         3
                                                             FILE
                                                          *
                                                                         : 2024-06-06↓
                                                             DATE
                                                         4
                                                          *
                                                             DESCRIPTION : Main Program↓
                                                           *
   extern void MainTask(void);
                                                         6 ★↓
                                                             NOTE: THIS IS A TYPICAL EXAMPLE. \downarrow
                                                           *
   void main(void);
                                                         8
                                                          *√
   void main(void)
                                                        10 #include "r_smc_entry.h"
                                                        11 \downarrow
    . . . .
                                                        12 extern void MainTask(void);
    MainTask();
                                                        13 ↓
                                                        14 void main(void);↓
                                                        15 J
                                                        16 void main(void)↓
                                                        17 {↓
                                                            MainTask();↓
                                                        18
                                                        19 } [EOF]
               Copyright 2024
                                                   emWinスタートアップ・設定ガイド
                                                                                             2024年
EmbITeK
                                                                                                   P.36
```

"QE for Display [RX]"を使用します

6月

オプション2 AppWizardなしでemWin APIで画面作成: スタートアップコード

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

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

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



325	/**************************************
326	$\star \downarrow$
327	* MainTask↓
328	$*/\downarrow$
329	void MainTask(void) {\
330	<pre>int ReactOnMove;↓</pre>
331	\downarrow
332	ReactOnMove = 1; \downarrow
333	#if GUI_SUPPORT_MEMDEV↓
334	//WM_SetCreateFlags(WM_CF_MEMDEV); \downarrow
335	#endif↓
336	GUI_Init();↓
337	WM_MULTIBUF_Enable(1); \downarrow
338	//↓
339	// Check if recommended memory for the sample is available \downarrow
340	//↓
341	\downarrow
342	<code>WM_SetCallback(WM_HBKWIN, _cbBk);</code>
343	while (1) { \downarrow
344	<pre>_ExecScratchPad();</pre>
345	}↓
346	\downarrow
- 1-	



Copyright 2024 EmbiTeK Co., Ltd. All rights reserved.

^{オプション2} emWin APIで画面作成:サンプルアプリケーション

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

emWin Tutorials

EmbITeK

Lownload executables of the emWin Tutorials package [zip archive]	🛓 Download 7z archive
Lownload source code of the emWin Tutorials package [zip archive]	🛓 Download 7z archive

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

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

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



● オンラインデモサンプル(emWin API用):

EmbiTeK Co., Ltd. All rights reserved.

https://wiki.segger.com/emWin_Examples

Copyright 2024



• RX65N Envision Kitの詳細

<u>https://www.renesas.com/jp/ja/products/microcontrollers-microprocessors/rx-32-bit-performance-efficiency-mcus/rx65n-envision-kit-rx65n-envision-kit</u>

- RX65N / RX72N Envision Kit用emWinスタートアップガイド (PDF) <u>https://www.embitek.co.jp/technote/emwin/StartupGuide_emWin_EnvisionKit.pdf</u>
- RXファミリ QE for Display [RX] アプリケーションノート <u>https://www.renesas.com/jp/ja/document/apn/rx-family-qe-display-rx-application-note</u>
- emWinユーザガイド(オンライン)
 <u>https://www.segger.com/doc/UM03001_emWin.html</u>
- AppWizardユーザガイド(オンライン)
 <u>https://www.segger.com/doc/UM03003_AppWizard.html</u>



ソースパッケージアップグレードのご案内

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

※ ライセンス製品価格・割引率は、サポート窓口対応や諸条件、 為替相場により提示価格が変動する場合があります。 ソースアップグレードのメリット:

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

お気軽に以下窓口へお問い合わせください。

株式会社エンビテック

代理店販売(デバッガ、RTOS、GUI、ミドルウエア)

TEL: 03-6240-2655 / FAX: 03-6240-2656
E-mail : sales@embitek.co.jp
https://www.embitek.co.jp





Copyright 2024 EmbiTeK Co., Ltd. All rights reserved.