

スタートアップガイド

更新日:2022年9月

RX65N / RX72N Envision キット用 emWin デモサンプル

本資料は「emWin for RX」OEM版GUIソフトウエアの導入及びデモサンプルの動作確認方法を解説します。



RX65N Envision Kit



RX72N Envision Kit

本資料に記載の全ての情報の使用に起因する損害、第三者の知的財産権、権利またはその他の経 費に対して、SEGGER 社及び株式会社エンビテックは一切責任を負いません。 本資料の内容は予告なく変更されることがあります。

商標

「EmblTeK」、EmblTeK ロゴは株式会社エンビテックの商標または登録商標です。その他、本資料に記載しているプロセッサ名、ツール名および製品名は、それぞれ各社の商標または登録商標です。





目 次

1. emWin のインストール	. 3
1.1. ダウンロード	3
1. 2. パッケージ内容	4
2. RX72N Envision Kit 用サンプル	6
2.1. デモサンプル	8
2.1.1. SlideDemo:マルチウインドウスライドデモアプリケーション	8
2.2. プロジェクトビルド方法(e2Studio)	9
2.3. ダウンロード・デバッグ方法	12
2.4. CS+ IDE 環境への移植方法	15
2. 4. 1. e2Studio プロジェクトを CS+ IDE にインポート	15
2.4.2. プロジェクト設定を修正	20
2.4.3. デバッグ設定及びダウンロード・デバッグ操作	23
3. RX65N Envision Kit 用サンプル	25
3.1. デモサンプル	27
3.1.1. 標準サンプルアプリケーション	27
3.1.2. マルチタッチ機能のデモアプリケーション	28
3.1.3. マルチウインドウスライドデモアプリケーション	28
3.2. プロジェクトビルド方法(e2Studio)	29
3.3. ダウンロード・デバッグ方法	32
3.4. CS+ IDE 環境への移植方法	35
3. 4. 1. e2Studio プロジェクトを CS+ IDE にインポート	35
3.4.2 プロジェクト設定を修正	40
3. 4. 3. デバッグ設定及びダウンロード・デバッグ操作	45



1. emWinのインストール

1.1. ダウンロード

RX65N/RX72 用 emWin プログラムを以下の URL からダウンロードしてください(要ユーザー登録)。

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

パッケージ内容は以下 URL のページをご参照ください。

https://www.segger.com/products/user-interface/emwin/add-ons/emwin-support-renesas-rx-mcu/





ダウンロードソフトウエアパッケージを、開発環境インストール済みのユーザーパソコンのローカルド ライブにコピーします。Windows インストーラは不要です。



- 📕 Doc
- > 📙 emWin_RX65N
- > 📕 emWin_RX72N
- >] emWin_Simulation
 - looT
- <install root>

	– Doc		ユーザーマニュアル・リリースノート
	- emWin_RX72N	•••••	emWin ライブラリ、RX72N Envision ボード用デモサンプル
	- emWin_RX65N	•••••	emWin ライブラリ、RX65N Envision ボード用デモサンプル
	– emWin_Simula	tion	emWin シミュレータプログラム・チュートリアル
l			パソコン用ツール(GUI ビルダ、フォント変換ツール等)

「emWin for RX」ライブラリのライセンス利用条件は「Doc」フォルダ下の「License.txt」ファイル内容を ご確認ください。

1.2. パッケージ内容

● RX用 GUI ライブラリ (バイナリ式):

- emWin Color basic Package
- ・ emWin GUIDRV Lin (ディスプレードライバサポート)
- ・ ウインドウマネジャー(すぐに使用可能な複数ウィジェットサポート)
- ・ メモリデバイス、アンチエイリアス機能サポート
- ・ VNC Server (GUI アプリケーションの遠隔操作が可能)
- ・ マルチタッチサポート

対象マイコンデバイス:RXマイコンシリーズ(RXv1、RXv2、RXv3)

● 開発支援ツール(パソコン用 EXE 式):

- ・ 画面作成・デザインツール (AppWizard、GUIBuilder)
- ・ フォント変換ツール
- ・ 画像データ変換ツール
- ・ JPEG→動画変換ツール
- ・ バイナリデータ変換ツール (Bin2C)
- Unicode データ変換ツール(U2C)
- ・ emWinシミュレータ

名前	サイズ	更新日時
🚳 AppWizardRenesas_V128b_626b_Install.exe	42,974 KB	2022/06/02 11:42
Bin2C.exe	104 KB	2022/06/02 11:42
BmpCvt_V626b_Install.exe	5,285 KB	2022/06/02 11:42
BmpCvtCL_V626b_Install.exe	5,225 KB	2022/06/02 11:42
👘 emVNC_V626_Install.exe	5,122 KB	2022/06/02 11:42
► emWinPlayer.exe	3,416 KB	2022/06/02 11:42
🝳 emWinSPY.exe	3,350 KB	2022/06/02 11:42
🥡 emWinView_V626_Install.exe	5,304 KB	2022/06/02 11:42
🔣 GUIBuilder.exe	1,336 KB	2022/06/02 11:42
HTTT: JPEG2Movie.exe	126 KB	2022/06/02 11:42
💑 License.txt	9 KB	2022/06/02 11:42
SetupFontCvt_V626.exe	430 KB	2022/06/02 11:42
U2C.exe	103 KB	2022/06/02 11:42

ツール	解説
AppWizardRenesas_Vxx_Install.exe	GUI デザイン・ビルダツール
Bin2C. exe	BIN データ→C 言語変換ツール
BmpCvt_Vxx_Install.exe	ビットマップ変換ツール
BmpCvtCL_Vxx_Install.exe	ビットマップ変換ツール(コマンドライン)
emVNC_Vxx_Install.exe	VNC クライアントプログラム
emWinPlayer.exe	EMF 式 emWin 動画の再生ツール
emWinView_Vxx_Install.exe	emWin シミュレータのデバッグ・ツール
JPEG2Movie.exe	JPEG→動画変換ツール(EMF 式動画を作成します)
SetupFontCvt_Vxx. exe	RX 環境用フォント変換ツール
U2C. exe	UTF8 データ→C 言語変換ツール

※パソコン側での GUI デザインは従来式の「GUI Builder. exe」ではなく、最新式の「AppWizard」デザインツールのご利用をお勧めいたします。

● BSP・アプリケーション・チュートリアル:

- ・ RX65N Envision Kit ボード用 GUI サンプル
- ・ RX72N Envision Kit ボード用 GUI サンプル
- ・ emWin シミュレータ用チュートリアルサンプル

RX65N Envision Kitの詳細:

https://www.renesas.com/jp/ja/products/software-tools/boards-and-kits/eval-kits/rx65n-envision-kit. html

RX72N Envision Kitの詳細:

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



2. RX72N Envision Kit 用サンプル

● RX72N 用 emWin プログラム

ダウンロードソフトウエアパッケージの「emWin_RX72N」フォルダをユーザーフォルダにコピーします。



< emWin_RX72N >

⊢ Application │ └ SlideDemo	マルチウインドウスライドデモアプリケーション
 - Config - GUI	GUI モジュール・LCD ドライバコンフィグレーション設定 GUI モジュールライブラリ及びヘッダ

• RX72N Envision ボードについて





	マイコンデバイス:	RX72N(ROM:4MB、内蔵 RAM:1MB)
	LCD パネル:	TFT WQVGA (480 x 272)
	デバッガ I/F:	オンボード E2–Lite デバッガ
%RX72N	Envision キットの詳しい仕	様については取扱説明書をご参照ください。

● RX72N Envision ボード設定



SW1-2	ON	USB シリアルモード エミュレータ・デバッガ使用不可
	OFF	プログラマ・デバッガモード エミュレータ・デバッガ使用可能

デバッグモードの設定: SW1-2: OFF



2.1. デモサンプル

2.1.1. SlideDemo:マルチウインドウスライドデモアプリケーション



● 開発環境

コンパイラ:	Renesas	RXCC コンノ	パイラ (v3	3. 00. 00 以降)
IDE:	Renesas	e ² Studio	(Version	2022-07 以降)
	CS+ for	CC (V8 以降	夅)	



2.2. プロジェクトビルド方法 (e2Studio)

● Step1:e²Studio IDE を起動してプロジェクトを開きます

[ファイル]→ [インポート]コマンドで emWin_RX72N フォルダ下のプロジェクトをワークスペースにインポートします。





📴 workspace - e² studio					- 0	×
ファイル(F) 編集(F) ソース(S) リファクタリング(T) ナビゲート(N) 検索(A) プロジェクト(P) Renesas Views 宇行(R) ウィン	ごかいの ヘルプ(円)				
		8:20 ± ▼ 0. ▼ 10. ▼ ±. m	▼ mm ¥ कि 201301		* * *	1 Re 101 - er
			100 H 10 10 1 W 1			
				く 田 地C/C++	× XY-F-J2.	1196-9
[▶] プロジェクト・エクスプローラー×				こ 日 話 アウトライン	/ ×	- 0
 BX72N_EnvisionKit (in emWin_RX72N) 				アウトラインを	提供するアクティ	ィブなエディ
> 🔊 Includes				ターはあります	さん。	
 Application 						
> 🔄 SlideDemo						
> 🧭 Config						
✓ [™] src						
Y 🗁 smc_gen						
> 🗁 general						
> 🗁 r_bsp						
> 🗁 r_cmt_rx						
> 🗁 r_config						
> General real control of the second seco						
> 🗁 r_drw2d_rx						
> 🗁 r_glcdc_rx						
> 🗁 r_gpio_rx	2 問題 × ロコンソール 、スマート・ブラウザー 、フ	マート・マニュアル			Я	7 8 - 0
> Ger_pincfg	0 errors, 1 warning, 0 others					
> 🗁 r_sci_iic_rx	記述/説明	リソース	パス	ロケーション	型	
> 🗁 r_sys_time_rx	> 💩 警告 (1 項目)					
RX72N_EnvisionKit.c						
> 🗁 GUI						
RX72N_EnvisionKit HardwareDebug.launch						
				スマート・マニュアル		×
				新しいスマート・マニュア	ルがインストールさ	れました:
				Renesas RX72N Sma	rt Manual Rev.	1.11

● Step2:プロジェクトビルド設定内容を確認します

プロジェクトプロパティダイアログを開いて、「C/C++ビルド → 設定」タブ下のビルド環境設定内容を確認しま す。インストールされている CC-RX ツールチェーンバージョンとサンプルプロジェクトの設定が異なる場合は、 CC-RX ツールチェーンバージョンを正しく設定します。

イルタ入力	設定		
>> リソ−ス > C/C++ ビルド スタック解析 ツ−ルチェイン・エ	Configuration: HardwareDebug [アクティブ]		
ビルド変数	⑧ ツール設定 Toolchain Device	Enable toolchain integration	
ロキンク環境	Enable toolchain integration	現在のツールチェーン	
設定	- 理なのツールチェーン	"- ILFT-""POPOSAS CCPY	
C/C++ 一般	ツールチェーン: Renesas CCRX	If Sight 20100	
Renesas QE	パージョン: v3.01.00	V-232: V3.01.00	
Task Tags Validation	ツールチェーンの変更 (タブを切り替える前に「適用」ボタンをクリックしてください)	ツールチェーンの変更 (タブを切り替える前に「適用」ボタン?	をクリックしてください)
ビルダー	ツールチェーン:	ツールチェーン: Renesas CCRX	~
プロジェクト・ネーチャ プロジェクト参照	パージョン:	バージョン: v3.04.00	~



Step3:プロジェクト(デバッグ構成)をビルドします

プロジェクトエクスプローラーウインドウのプロジェクト名を選択し、メニューから「プロジェクト → プロジェクトのビルド」又は(CtrI+B)コマンドを実行してプロジェクトを再ビルドします。

🔄 workspace - e² studio



ビルド後に「HardwareDebug」フォルダ下に「RX72N_EnvisionKit.x」デバッグ式 ELF イメージが作成されます。



2.3. ダウンロード・デバッグ方法

Renesas e²Studio IDE 環境でのフラッシュダウンロード・デバッグ操作にデバッグ構成が必要です。 RX72N Envision キット用サンプルに「RX72N_EnvisionKit HardwareDebug」デバッグ構成が含まれています。 以下の 2 種類のデバッグ設定方法を解説します。

① 「RX72N_EnvisionKit HardwareDebug」デバッグ構成を使用してデバッグ

② 新規デバッグ構成を作成して行うデバッグ操作

● Step1:ボードをデバッグモードに設定します

ボード上の SW1 DIP-SW を以下のように設定します。

```
SW1-2:0FF
```

キット付属ケーブルでボード上のmini-USB コネクタ(ECN1)をパソコンに接続します。

● Step2:「RX72N_EnvisionKit HardwareDebug」デバッグ構成を確認します

workspace - e* studio	
ファイル(F) 編集(E) ソース(S) リファクタリン	/グ(T) ナビゲート(N) 検索(A) プロジェクト(P) Renesas Views 実行(F
🐔 🚁 🔳 🎋 デバッグ(B) 🛛 🖉 RJ	X72N_EnvisionKit Har 🗸 🌣 🛄 🗨 🗟 🐚 🕲 🔻 🗞 🖛 📓 🗳 🧌
월 * 월 * 10 라 10 * 0 * 11 1	
」 プロジェクト・エクスプロ−ラ− ×	🖪 🔩 🁔 📲 🗖 🧪 編集アイコンを
🖉 🐼 RX72N_EnvisionKit (in emWin_RX	(72N) [HardwareDebug] クリック
> 🐝 パイナリー	
Edit Renesas GDB Hardware Debugging co aunch Configuration Name: スイン 参 Debugger Startup シノス □ 共通(C	onfiguration RX72N_EnvisionKit HardwareDebug for デバッグ(D) HardwareDebug
Edit Renesas GDB Hardware Debugging co aunch Configuration Name: RX72N_EnvisionKit H コメイン 参 Debugger M Startup シソース 回 共通(C プロジェクト(P):	onfiguration RX72N_EnvisionKit HardwareDebug for デバッグ(D) HardwareDebug
Edit Renesas GDB Hardware Debugging co aunch Configuration Name: スイン 参 Debugger シ Startup シース ゴ 共通(C プロジェクト(P): RX72N_EnvisionKit	onfiguration RX72N_EnvisionKit HardwareDebug for デバッグ(D) HardwareDebug () 参照(B)
Edit Renesas GDB Hardware Debugging co aunch Configuration Name: RX72N_EnvisionKit H シオイン 参 Debugger M Startup シノス 四 共通(C プロジェクト(P): RX72N_EnvisionKit C/C++ アプリケーション:	onfiguration RX72N_EnvisionKit HardwareDebug for デバッグ(D) HardwareDebug
Edit Renesas GDB Hardware Debugging co aunch Configuration Name: RX72N_EnvisionKit F シイン 参 Debugger ● Startup シノース □ 共通(C プロジェクト(P): RX72N_EnvisionKit C/C++ アプリケーション: HardwareDebug/RX72N_EnvisionKit.x	onfiguration RX72N_EnvisionKit HardwareDebug for デバッグ(D) HardwareDebug
Edit Renesas GDB Hardware Debugging co aunch Configuration Name: RX72N_EnvisionKit ト ンイン や Debugger > Startup シノース 回 共通(C プロジェクト(P): RX72N_EnvisionKit C/C++ アプリケーション: HardwareDebug/RX72N_EnvisionKit.x	onfiguration RX72N_EnvisionKit HardwareDebug for デバッグ(D) HardwareDebug () () を照(B) を照(B) を照(B)
Edit Renesas GDB Hardware Debugging co aunch Configuration Name: RX72N_EnvisionKit F メイン 参 Debugger ● Startup ● ソース □ 共通(C プロジェクト(P): RX72N_EnvisionKit C/C++ アプリケーション: HardwareDebug/RX72N_EnvisionKit.x 起動前に必要に応じてビルド	onfiguration RX72N_EnvisionKit HardwareDebug for デバッグ(D) HardwareDebug C) 変数(V) プロジェクトの検索(H) 参照(R)
Edit Renesas GDB Hardware Debugging co Launch Configuration Name: RX72N_EnvisionKit F メイン 参 Debugger ● Startup ● ソース □ 共通(C プロジェクト(P): RX72N_EnvisionKit C/C++ アプリケーション: HardwareDebug/RX72N_EnvisionKit.x 起動前に必要に応じてビルド Build Configuration: Select Automatically	onfiguration RX72N_EnvisionKit HardwareDebug for デバッグ(D)
Edit Renesas GDB Hardware Debugging co Launch Configuration Name: RX72N_EnvisionKit F メイン 参 Debugger ● Startup ● ソース □ 共通(C プロジェクト(P): RX72N_EnvisionKit C/C++ アプリケーション: HardwareDebug/RX72N_EnvisionKit.x 起動前に必要に応じてビルド Build Configuration: Select Automatically ○ 自動ビルドを有効にする	onfiguration RX72N_EnvisionKit HardwareDebug for デバッグ(D) HardwareDebug



🖪 構成の編集	
Edit Renesas GDB Hardware Debugging configurat	ion RX72N_EnvisionKit HardwareDebug for デバッグ(D)
Launch Configuration Name: RX72N_EnvisionKit HardwareDe	ebug
■メイン 参 Debugger ► Startup をソース □ 共通(C)	
Debug hardware: E2 Lite (RX) Y Target Device: R5F	572NN
GDB Settings Connection Settings デバッグ・ツール設定	
✓ クロック	
メイン・クロック・ソース	EXTAL
EXTAL 周波数[MHz]	16.0000
動作周波数 [MHz]	
内部フラッシュメモリー書き換え時にクロック・ソースの変更を調	许可する はい 🗸
∨ ターゲット・ボードとの接続	
エミュレーター	(Auto)
接続タイプ	Fine
JTag クロック周波数[MHz]	6.00
Fine ボーレート[Mbps]	1.50
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
✓ CPU 動作モード	
レジスター設定	シングルチップ
モード端子	シングルチップ・モード
起動バンクを変更する	いいえ
起動バンク	パンクロー・
◇ 通信モード	
モード	デパッグ・モード
デパッガー終了後にユーザー・プログラムを実行する	いいえ
▼ フラッシュ	
ID J-F	FFFFFFFFFFFFFFFFFFFFFFFFF

▶ Step3:デバッグセッションを開始します

📴 workspace - e² studio









プロジェクトを実行してデモサンプルを動かして動作確認を行います。



2.4. CS+ IDE 環境への移植方法

2. 4. 1. e2Studio プロジェクトを CS+ IDE にインポート

● Step1:e²Studio IDE を起動してプロジェクトを開きます

[ファイル]→ [インポート]コマンドで emWin_RX72N フォルダ下のプロジェクトをワークスペースにインポート

します。

workspace - e² studio

新規(N) Alt+シフト+N> ファイルを開く() □ ファイル・システムからプロジェクトを開く 最近のファイル >	 ゴンボート 選択 	 2
閉じる(C) Ctrl+W すべて閉じる(L) Ctrl+シフト+W	アーカイブ・ファイルまたはディレクトリーから新規プロジェクトを作成します。	
 保存(5) Ctrl+S 名前を付けて保存(A) すべて保存(E) Ctrl+シフト+S 前回保存した状態に戻す(T) 	インポート・ウィザードの選択(S): フィルタ入力 ♪ アーカイブ・ファイル	
移動(V) 名前を変更(M) F2 更新(F) F5 行区切り文字の変換(D) >	 □ ファイル・システム □ フォルアーよたはアーカイプ由未のプロジェクト ② 既存プロジェクトをワークスペースへ □ 設定 > ▷ C/C++ 	
⇒ インボート(I)	> 🖻 Oomph	
プロパティ(R) Alt+Enter ワークスペースの切り替え(W) >	↓	
再開 終了(X)	⑦ < 戻る(B) 次へ(N) > 終了(F)	キャンセル
インボート プロジェクトをインボート 既存の Eclipse プロジェクトを検索するディレクトリーを選択します。	- □ ×	
 インボート プロジェクトをインボート 既存の Eclipse プロジェクトを検索するディレクトリーを選択します。 ルート・ディレクトリーの選択(①: C:¥Workspace¥RX¥emWin- 	- □ × 。 -RX-220603¥emWin_RX72N ~ 参照(<u>R</u>)	
 インボート プロジェクトをインボート 既存の Eclipse プロジェクトを検索するディレクトリーを選択します。 ルート・ディレクトリーの選択(①: C:¥Workspace¥RX¥emWin- アーカイブ・ファイルの選択(Δ): プロジェクト(P): 	- □ × 	
 インボート プロジェクトをインボート 既存の Eclipse プロジェクトを検索するディレクトリーを選択します。 ルート・ディレクトリーの選択(①: C:¥Workspace¥RX¥emWin- アーカイブ・ファイルの選択(Δ): プロジェクト(D): 図RX72N_EnvisionKit(C:¥Workspace¥RX¥emWin-RX-2 	- ロ × -RX-220603¥emWin_RX72N ~ 参照(B) 220603¥emWin_RX72N) すべて選択(S)	
 「コジェクトをインボート 既存の Eclipse プロジェクトを検索するディレクトリーを選択します。 ルート・ディレクトリーの選択(①: C:¥Workspace¥RX¥emWin- アーカイブ・ファイルの選択(Δ): プロジェクト(D):	× -RX-220603¥emWin_RX72N ~ 参照(R) 220603¥emWin_RX72N) すべて選択(S) 選択をすべて解除(D) 更新(C)	
 	- RX-220603¥emWin_RX72N - RX-220603¥emWin_RX72N 220603¥emWin_RX72N) ダ照(B) 参照(C) 運択をすべて解除(D) 更新(E)	
 [●] インボート 現存の Eclipse プロジェクトを検索するディレクトリーを選択します。 ・ルート・ディレクトリーの選択(①: C:¥Workspace¥RX¥emWin- ○ アーカイブ・ファイルの選択(Δ): 「ロジェクト(P): 「ロジェクト(P): 「図X72N_EnvisionKit(C:¥Workspace¥RX¥emWin-RX-2	- RX-220603¥emWin_RX72N 参照(D) 参照(D) 多照(D) 多照(D) 多照(D) 多照(D) 多照(D) 多照(D) 多照(D) 多照(D) 多照(D)	
 		
 	- RX-220603¥emWin_RX72N - RX-220603¥emWin_RX72N 220603¥emWin_RX72N) タベて選択(5) 選択をすべて解除(D) 更新(E) 新規(W) 選択(E)	



🔄 workspace - e² studio					- 0	×
ファイル(F) 編集(E) ソース(S) リファクタリング(T) ナビゲート(N)	検索(A) プロジェクト(P) Renesas Views 実行(R) ウ	ィンドウ(W) ヘルプ(H)				
	ar * 🔅 i 🗂 🕶 📾 🗑 🕶 🐐 🖷 🖼 🐂 🕷 👘	: @ <mark>\$ \$ • 9</mark> • 6, • \$ 0	- m k S 2 0 1 1 0 1	• 63 • 6 • G	- 🦢 🛷 -	1 IV III 1
N + N + 6 + 6 + 0 + 10			Q	1 B C/C++	スマート・コン	イグレータ
■ プロジェクト・エクスプローラー×			-	D - 701-547	×	
× BX72N EnvisionKit (in emWin RX72N)				アウトライン大き	見供オスマカテ	(ゴたエディ
> 🔊 Includes				ターはありませ	Engel J J J J	1/41/1
 Application 						
> 🕞 SlideDemo						
> 🐸 Config						
Y 🐸 src						
✓ ➢ smc_gen						
> 🗁 general						
> 👺 r_bsp						
> 🗁 r_cmt_rx						
is r_config						
> is r_dmaca_rx						
> ≥ r_drw2d_rx						
> 🗁 r_glcdc_rx						
> is r_gpio_rx	■ 問題× □ コンソール ● スマート・ブラウザー	177 L 7-27				
		** YA-L.A-T.W				8 - 0
> 🗁 r_pincfg	0 errors, 1 warning, 0 others	~X4-L.4-T.W			ម	8 - 0
> 🗁 r_pincfg > 🗁 r_sci_lic_rx	0 errors, 1 warning, 0 others 記述/説明	Jy-2	パス	ロケーション	型	8 - 0
<pre>> extbf r_pincfg > extbf r_sci_lic_rx > extbf r_sci_lic_rx > extbf r_sys_time_rx</pre>	0 errors, 1 warning, 0 others 記述/説明 > ▲ 警告 (1 項目)	ערב-ארא	パス	ロケーション	9 型	0
 >>>> cr_pincfg >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	0 errors, 1 warning, 0 others 記述/説明 > ▲ 警告 (1 項目)	ערביאר איז	152	ロケーション	9 型	
 >>> r_pincfg >>>> r_sci_iic_rx >>>>>>>>> r_sys_time_rx >>>> >>>>>>>>>>>>>> >>>>>>>>>>>> >>>>>>>>>>>>>>>>> >>>>>>>>>>>>>>>>>> >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	0 errors, 1 warning, 0 others 記述/説明 > ▲ 警告 (1 項目)	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	152	ロケーション	型	
 > Image: provide the second sec	0 errors, 1 warning, 0 others 記述/説明 > ▲ 警告 (1 項目)	ערב-יז-אע	1(2	ロケーション	型 型	
 > Image: provide the second sec	0 errors, 1 warning, 0 others 記述/説明 > ▲ 警告 (1 项目)	ערב-יז-א	1(2	ロケーション	9 10 10 10 10 10 10 10 10 10 10 10 10 10	8 - 0
 > Image: provide the second sec	0 errors, 1 warning, 0 others 記述/説明 > ▲ 警告 (1 项目)	X-rv	1(2	ロケーション	9 型	.8 = 0
 > ar_pincfg > ar_psci_lic_rx > ar_psc_stime_rx > aRX72N_EnvisionKit.c > and argument of the state of the stat	0 errors, 1 warning, 0 others 記述/説明 > ▲ 警告 (1 项目)	ערבייי-גייע	112	ロケーション -ト・マニュアル	型 型	×

● Step2:プロジェクトを RCPC ファイルとしてエクスポートします

「RX72N_EnvisionKit」プロジェクトをアクティブモードに設定します。

[ファイル]→ [エクスポート]コマンドで RX72N_EnvisionKit プロジェクトを CS+ 以降と互換性がある RCPC 式 ファイルにエクスポートします。





「一般 → Renesas 共通プロジェクト・ファイル」を選択します。

エクスポート	-	×		
選択 CS+のインボート用にルネサス共通プロジェクト・ファイル形式でプロジェクトをエクスボートし	ます。	4		
エクスポート・ウィザードの選択(S):				
71ルタ入力				
✓ ≥ 一般 ≥ Renesas FSP Archive File		^		
◎ Renesas FSP User Pack Renesas共通プロジェクト・ファイル P. アーカイブ・ファイル		~		
②< 戻る(B)次へ(N) >終了(F)	キヤン	セル		
			-	
🗐 プロジェクトのエクスポート				×
共通プロジェクト・ファイルにエクスポート				
選択した e2 studio プロジェクトを、共通プロジェクト・ファイル・フォーマット す。	V1.0 にエクスオ	∛−トしま		
選択されたプロジェクト:				
共通プロジェクト・ファイルのデフォルトのロケーションは、選択したプロジェクトの)ルート・ディレク	トリーです。		
CS+にインボートする時は、先に RX72N_EnvisionKit.rcpc を更新してくださ	L1.			
U7->32: C:/Workspace/RX/emWin-RX-220603/emWin_RX72N	(RX/2N_Envis	ionKit.rcp	c	!参照!
⑦ < 戻る(B) 次へ(N) >	終了(F)		キャンセ	JI-

エクスポート操作後に「emWin_RX72N」フォルダ下に「RX72N_EnvisionKit.rcpc」プロジェクトファイルが作成 されます。

✓	.settings		2022/09/22 16:35	ファイル フォルダー
✓	Application		2022/09/22 16:35	ファイル フォルダー
> 📜 .settings	📙 Config		2022/09/22 16:35	ファイル フォルダー
🗸 📙 Application	📕 GUI		2022/09/22 16:35	ファイル フォルダー
📕 SlideDemo	📕 src		2022/09/22 16:35	ファイル フォルダー
Config	IDE .cproject	39 KB	2020/08/06 10:45	CPROJECT ファイル
	IDE .project	1 KB	2020/04/06 20:42	PROJECT ファイル
GUI	RX72N_EnvisionKit HardwareDebug.launch	10 KB	2020/08/06 10:44	LAUNCH ファイル
🗸 📙 src	RX72N_EnvisionKit.rcpc	37 KB	2022/09/23 15:43	RCPC ファイル
> 📜 smc_gen				



● Step3:プロジェクトを閉じて e2Studio を終了します

📴 workspace - e² studio

ファイル(F) 編集(E) ソース(S) リファクタリング(T) ナビゲート(N) 検索(A) プロジェクト(P) Renesas Views 実行(R) ウィンドウ(W) ヘルプ(H)

6	🌜 🐲 🔳 🏘 デパッグ(B) 🚽 🖻 RX72N_EnvisionKit Har Y 🍫		プロジェクトを聞く(E)		\$ ▼ Q ▼ 0 ₀ ▼ 株 m
1 18			プロジェクトを閉じる(S)		
٩.	プロジェクト・エクスプローラー × 🖻 😘 🎖 🕴 🗖	-	Open FSP Configuration		
*	 RX72N_EnvisionKit (in emWin_RX72N) [HardwareDebug] Includes Application 		すべてビルド(A) ビルド構成 プロジェクトのビルド(B)	Ctrl+Alt+B > Ctrl+B	
			ワーキング・セットのビルド(W) クリーン(N)	>	

● Step4:プロジェクトを CS+環境で開きます

開発環境: CS+ for CC V8.08.00

CS+ IDE を起動して、[プロジェクト] → [プロジェクトを聞く..]コマンドを事項し、Step2 で作成されました e2Studio エクスポートプロジェクトファイル「RX72N_EnvisionKit.rcpc」を開きます。

🚺 CS+ for CC - [スタート]				
ファイル(F) 編集(E) 表示(V) プ	ロジェクト(P) ビルド(B) デバッグ(D) ツール(T) ウイン	ィドウ(W) ヘルプ(H)	
🎧 スタート(S) 🛛 🔒 🗐 🌆	新しいプロジェクトを作成(N)		00% 👻 🖬 🖬	- 1
ାଟଟ <i>⊡</i> ଅଷ୍ଟା⊡ ‡ ■	プロジェクトを開く(P)			
プロジェクト・ツリー 🛛 🕂 🕻	お気に入りのプロジェクト(F)	+		
2 🕜 🙎 🛢	追加(D)	•		
			4	

🚳 プロジェクトを開く		×	
\leftrightarrow \rightarrow \checkmark \uparrow \blacksquare \ll RX \rightarrow emWin-RX-220603	→ emWin_RX72N → v Ō	,○ emWin_RX72Nの検索	
整理 ▼ 新しいフォルダー		≣≕ ▾ Ⅲ ?	
CS+ ^	名前 ^	サイズ 更新日時	
emWin-RX-220603	.settings	2022/09/22 16:35	
emWin_RX72N	Application	2022/09/22 16:35	
HardwareDebug	Config	2022/09/22 16:35	
LiPP_RX65N	GUI	2022/09/22 16:35	
PS_FS_RX651	src	2022/09/22 16:35	
PS_FS_RX651_32BPP	RX72N_EnvisionKit.rcpc	37 KB 2022/09/23 15:43	
QE_for_Display_sample_RX65N_Envis			
OEDisp RX72N Envision	<	>	
ファイル名(N): RX72N_Envisio	nKit.rcpc	✓ MCU Simulator Online ∕ e ² stud ✓	
		7031271-77-(1) (timep)	
		MCU Simulator Online/e [*] studio用フロジェクト・ファイル (*.rcpc) CubeSuite田プロジェクト・ファイル (*.cspi)	
		HEW用ワークスペース・ファイル (*.hws)	
		HEW用フロジェクト・ファイル (*.hwp) PM+田ワークスペース・ファイル (*.ppw)	
		PM+用プロジェクト・ファイル (*.prj)	



● Step5:プロジェクトマイコンデバイス設定を確認します

1			
プロジェクト変換設定		×	
プロジェクト(P):			
? 説明 ℝX72N_EnvisionKit	プロジェクト設定 変換先マイクロコントローラ マイクロコントローラ(T): 使用するマイクロコントローラを検 ● R5F572NNH×F ● R5F572NNH×F ■ R5F577NNH×F ■ R5F577NNH×F ■ R5F577NNH×F ■ R5F577NNH×F ■ R5F577NNH×F ■ R5F577NNH×F ■ R5F577NH×F ■ R5F577NH×F ■ R5F577NH×F ■ R5F577NH×F ■ R5F577	RX (M): 素できます) B(144pin) B(144pin) B(144pin) B(144pin) B(144pin) B(144pin) B(144pin) B(144pin) B(144pin) B(144pin) B(144pin) B(144pin) C(176pin) C(176pin) C(176pin) C(176pin) C(176pin) C(176pin) C(176pin) C(176pin) C(176pin) C(176pin) C(100pin) P_DUAL(190pi C(100pin) P_DUAL(190pi C(100pin) P_DUAL(190pi C(100pin) P_DUAL(190pi C(100pin) C(100pin) P_DUAL(190pi C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin) C(100pin)	>
	プロジェクトの種類(K):	空のアプリケーション(CC-RX)	
	プロジェクト名(N):	RX72N_EnvisionKit	
	□ 変換元プロジェクトと異なる	る場所に作成する(A)	
	作成場所(L):	C:¥Users¥santo¥OneDrive¥ドキュメント¥CS+ Pr 🝺 🗸 参照(R)	
	☑ 変換元プロジェクト・フ		
	☑ 変換直後のブロジェクト構成	或ファイルをまとめてバックアップする(C)	
		OK キャンセル ヘルプ(H)	
🛞 RX72N_EnvisionKit - CS+ for CC - [プロパティ]		_	
ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B)	デバッグ(D) ツール(T) ウインドウ(W) ヘルプ(H)		🤤 🎯 🖨
🚳 79-F(S) 🔒 🖶 🗿 🕺 🖓 🖻		📉 🤅 ன 🕅 HardwareDebug 🔹 🖌 🤅 🖓 🗅 🤲 🔘 🕞 🕞 🧠 🎯 🖙 🖓 🖕	
: 💱 💱 🕍 🧐 🥵 : 🔲 💭 🗣 🔍 🗗 🤅 🖉	ソリューション一覧(S)		





2.4.2. プロジェクト設定を修正

● Step6:コンパイル・オプション設定を修正します



コピーして新しいパスを設定します。





● Step7:リンク・オプション設定を修正します

。 〇 RX72N_EnvisionKit - CS+ for CC - [プロパティ]		
ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デ	バッグ(D) ツール(T) ウインドウ(W) ヘルプ(H)	🤤 🍘 👜
	🔹 🔋 🖓 🖓 HardwareDebug 🔹 🍹 🖓 🕒	
🖓 🖓 🐙 🧖 🔲 📮 📮 👰 🗗 🎽 ソリューション 💷 プロジェクト・シリー 🛛 🗣 🗮	- 특(S) 같	パスを設定

コピーして新しいパスを設定します。

パス編集	×	
パス(1行につき1つのパス)(P):	R	%ProjectDir%/GUI/GUILib.lib
%ProjectFolder%/./././U	sers/santo/e2_studio/workspace/%P_lient	
	×	
<	>	
参照(B)		
│ □ 存在しないパスを許可する	5(N)	
プレースホルダ(L):		
プレースホルダ	値 ^	
ActiveProjectDir	C:#Workspace#RX#emWin-RX-220603¥em	
ActiveProjectMicomName ActiveProjectName	R5F572NNHxFB RX72N EnvisionKit	
BuildModeName	HardwareDebug	
MainProjectDir	C:¥Workspace¥RX¥emWin-RX-220603¥en 🗸	
<	>	
	OK キャンセル ヘルプ(H)	



● Step8:プロジェクトを保存→ビルドして設定を確認します

「プロジェクト→プロジェクトを保存」コマンドでプロジェクトを保存して、プロジェクトをビルドします。

٢	◎ RX72N_EnvisionKit - CS+ for CC - [プロジェクト・ツリー]				
フ:	アイル(F) 編集(E) 表示(V)	プロ	コジェクト(P) ビルド(B) デバッグ(D) ツール(T) ウインドウ(W) ヘレプ(H)	
8	🕅 スタート(s) 🛄 🔡 🎽	ß	新しいプロジェクトを作成(N)	😽 HardwareDebug	
19		-	プロジェクトを開く(P)		
-	プロジェクト・ツリー	1	お気に入りのプロジェクト(F) ・		
ĸ	2 🕜 🙎 🔳		追加(D)		
Ĩ	RX72N EnvisionKit	15			
- 그		8-// 12	依存プロジェクト設定(J)	ハ・モジュール・ファイル(0) 5ライブラリ・ファイル[1] ライブラリ・ファイル[0]	
アル	- 🔨 CC-RX (ビルド・ツ	1	プロジェクトを閉じる(E)	データ・ファイル[0]	
М	→ RX シミュレータ (R	プロジェクトを保存(R) Ctrl+Shift+S		
	□ ① ファイル	ABIC	名前を付けてプロジェクトを保存(T)	₩ 	

🕼 RX72N_EnvisionKit - CS+ for CC - [プロジェクト・ツリー]						
ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D) ツール(T) ウインドウ(W) ヘルプ(H)						
🙉 スタート(s) 🚽 📑 🎽 🐰		 HardwareDebug 				
🖓 ଟ୍ 🖉 🧐 🥵 🗇 📮 🤉	🗗 💋 ソリューション一覧(S) 🛔					
プロジェクト・ツリー	9 🗶 🔂 🖓 บาทรา					
الله الله الله الله الله الله الله الل	RX72N_E	EnvisionKit のプロパティ				
 ペンパン RX72N EnvisionKit () ペンパン RX5572NNHxFB (マイン・ ペンパン スマート・コンフィグレーク ペンパン パンパン マーク ペンパン パンパン マーク ペンパン アーク <l< td=""><td> RX72N_EnvisionKit をビルド(U) RX72N_EnvisionKit をリビルド(E) RX72N_EnvisionKit をクリーン(L) エクスプローラでフォルダを開く(F) Windowsエクスプローラのメニュー(</td><td>sionKit.mtpj s¥RR¥REL¥2020¥2010-emW (W)</td></l<>	 RX72N_EnvisionKit をビルド(U) RX72N_EnvisionKit をリビルド(E) RX72N_EnvisionKit をクリーン(L) エクスプローラでフォルダを開く(F) Windowsエクスプローラのメニュー(sionKit.mtpj s¥RR¥REL¥2020¥2010-emW (W)				

出力	џ	×
>src¥smc_gen¥r_bsp¥mcu¥all¥sbrk.c↓ >src¥smc_gen¥r_drw2d_rx¥src¥tes¥dave_pattern.c↓		^
>src¥smc_gen¥r_drw2d_rx¥src¥tes¥dave_triangle.cl		
>src#smc_gen¥r_gpio_rx#src¥r_gpio_rx.c. >src#smc_gen¥r_drw2d_rx#src#tes#dave_edge_c.		
Src#smc_gen¥r_dmaca_rx¥src¥r_dmaca_rx.c.]		
VRV2N_EnvisionKit.abs_ Benerse John John John John John John John John		
neresas declamizing Enner dompretedu ビルド終了(エラー:0個,警告:5個)(RX72N EnvisionKit, HardwareDebug)」 		
		¥
すべてのメッセージ *ビルド・ツール *ラピッド・ビルド		•
🔜 出力 🖗 スマート・ブラウザー 题 エラー一覧		





2.4.3. デバッグ設定及びダウンロード・デバッグ操作

● Step9:E2 Lite デバッグ・ツールを設定します

[デバッグ]→ [使用するデバッグ・ツール]コマンドで「RX E2 Lite」を選択します。

٢	RX72N_EnvisionKit - CS+ for CC - [プロジェクト・ツリー]					
フ	ァイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B)	デノ	「ッグ(D) ツール(T) ウインドウ(W) へ	レプ(H)			
8	🏽 🖧 🖓 🖓 🕄 📓 🦉 🕹 🖏 🖓 🖓	D,	デバッグ・ツールヘダウンロード(D)			- 🗒 🗛 🗠 i 🗩 i	
1	🖓 🖓 🖉 🧐 🔲 📮 🔍 🗗 🎽 🌈 ソリューショ	5	ビルド&デバッグ・ツールヘダウンロード	(B) F6			
-	プロジェクト・ツリー 🛛 🕴	5	リビルド&デバッグ・ツールヘダウンロー	۴(W)			
بر	2 🕜 🙎 🔳	00	デバッグ・ツールへ接続(C)				
Ţ		ď	デバッグ・ツールからアップロード(U)				
Ŕ		X		Shift+F6	-		
1.1	◎ フラート・コンフィグレータ(認計)/////)						
5							
ェアル			使用するデバッグ・ツール(L)	2		RX E2(2)	
ェアル			使用するデバッグ・ツール(L) 停止(S)	Shift+F5		RX E2(2) RX E2 Lite(E)	
2.アル			使用するデバッグ・ツール(L) 停止(S) 実行(G)	Shift+F5 F5		RX E2(2) RX E2 Lite(E) RX E1(Serial)(L)	
27JU			使用するデバッグ・ツール(L) 停止(S) 実行(G) ブレークサポに実行(E)	Shift+F5 F5		RX E2(2) RX E2 Lite(E) RX E1(Serial)(L) RX E1(ITAC)(C)	
2.77.Jb	 ▲ スマード・コンフィクレーダ(mg f ワール) ▲ CC-RX (ビルド・ツール) ▲ RX シミュレータ (デバッグ・ツール) ④ プログラム解析 (解析ツール) ● ③ ファイル ● ④ ビルド・ツール生成ファイル 		使用するデバッグ・ツール(L) 停止(S) 実行(G) ブレークせずに実行(E)	Shift+F5 F5 F8		RX E2(2) RX E2 Lite(E) RX E1(Serial)(L) RX E1(JTAG)(G)	
±77μ	 ▲ スマード・コンフィッレーッ(まえ= マール) ▲ CC-RX (ビルド・ツール) ▲ RX シミュレータ (デバッグ・ツール) ④ プログラム解析 (解析ツール) ● ① ファイル ● ① ビルド・ツール生成ファイル ● ① Application 		使用するデバッグ・ツール(L) 停止(S) 実行(G) ブレークせずに実行(E) ステップ・イン(I)	Shift+F5 F5 F8 F11		RX E2(2) RX E2 Lite(E) RX E1(Serial)(L) RX E1(JTAG)(G) RX E20(Serial)(R)	
2 アル	 ▲ スペード・コンフィッレーッ (mg = 19-20) ▲ CC-RX (ビルド・ツール) ▲ RX シミュレータ (デバッグ・ツール) ④ プログラム解析 (解析ツール) ● プログラム解析 (解析ツール) 		使用するデバッグ・ツール(L) 停止(S) 実行(G) ブレークせずに実行(E) ステップ・イン(I) ステップ・オーバー(O)	Shift+F5 F5 F8 F11 F10		RX E2(2) RX E2 Lite(E) RX E1(Serial)(L) RX E1(JTAG)(G) RX E20(Serial)(R) RX E20(JTAG)(T)	
27Jb	 ▲ スペード・コンフィッレーッ(modeling) ▲ CC-RX (ビルド・ツール) ▲ RX シミュレータ (デバッグ・ツール) ④ プログラム解析 (解析ツール) ● ③ ファイル ● ④ ビルド・ツール生成ファイル ● ● SlideDemo ● ● Config 		使用するデバッグ・ツール(L) 停止(S) 実行(G) ブレークせずに実行(E) ステップ・イン(I) ステップ・オーバー(O)	Shift+F5 F5 F8 F11 F10		RX E2(2) RX E2 Lite(E) RX E1(Serial)(L) RX E1(JTAG)(G) RX E20(Serial)(R) RX E20(JTAG)(T)	
277/b	 ▲ スペード・コンフィッレーッ (mg = 19-20) ▲ CC-RX (ビルド・ツール) ▲ マシミュレータ (デバッグ・ツール) ④ プログラム解析 (解析ツール) ● プログラム解析 (解析ツール) ● プログラム解析 (解析ツール) ● ③ ファイル ● ● ULド・ツール生成ファイル ● ● SlideDemo ● ● GUI 		使用するデバッグ・ツール(L) 停止(S) 実行(G) ブレークせずに実行(E) ステップ・イン(I) ステップ・オーバー(O) リターン・アウト(R)	Shift+F5 F5 F8 F11 F10 Shift+F11		RX E2(2) RX E2 Lite(E) RX E1(Serial)(L) RX E1(JTAG)(G) RX E20(Serial)(R) RX E20(JTAG)(T) RX シミュレータ(S)	

「RX E2 Lite」デバッグ・ツールの「接続用設定」タブの以下の項目設定を変更します。

メイン・クロック	16. 000
通信方式	FINE

วือวัรวระวงการสา 🗙		- x
2 🕜 🙎 🗷		P -+
<u> </u>	⊿ 内蔵ROM/RAM	*
	内蔵ROMサイズ[Kパイト] 4096	
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□	内蔵RAMサイズ[Kバイト] 1024 データフラッシュ・ソエリ・サイブ[レパイト] 92	
へCC-RX (ビルド・ツール)	Δ ΔΠ» Δ	
	メイン・クロック・ソース EXTAL	E
	メイン・クロック周波数[MHz] 16.0000	
	動作周波数[MHz]	
□□□ ファイル	内蔵フラッシュ・メモリ書き換え時のクロック操作を許可する いいえ	
↓ 1 ビルド・ツール牛成ファイル	▲ エミュレータとの接続	
	IミュレータシリアルNo.	
	▲ ダーノット・ホート(の)接続 エン-レークねこ示源(#\\\khttp://=	
i SlideDemo	ユミュレーシルの电源供給をする(取入200mA) いいえ 通信方式 ENJE	
🗄 🖬 Config		
GUI GUI	יואבוא דירוסייסייסייסייסייסייסייסייסייסייסייסייסיי	+
and arc	メイン・クロック周波数[MHz]	
	メイン・クロックがEXTALの場合、EXTAL周波数値を0.0001-99.9999の間で入力してください。	
KX/2N_EnvisionKit.c		
i smc_gen		
■ ■ プロジェクト変換情報_RX72N_EnvisionKit.txt	★ 毎年の日本では、「「「」」」、「「「」」、「「」」、「「」」、「「」」、「」、「」、「」、「」	-



• Step10:ダウンロードしてデバッグします

RX72N Envision 評価ボードの USB デバッグポートをパソコンに接続します。 [デバッグ]→ [ビルド&デバッグ・ツールへダウンロード]コマンドでプログラムをターゲットフラッシュにロ ードしてデバッグステップ・実行操作を確認します。





3. RX65N Envision Kit 用サンプル

● RX65N用 emWin プログラム

ダウンロードソフトウエアパッケージの「emWin_RX65N」フォルダをユーザーフォルダにコピーします。

🛛 퉬 emWin_RX65N	*	名前	更新日時	種類
> 퉲 .settings		Passauras	2010/10/14 15:24	
Application		ADD. BankSwan c	2019/10/14 15:24	C source file
			2017/10/20 14.40	C source file
MT ScratchPad		APP_BouncingBalls_AdvanceSim h	2017/12/05 16:02	C beader file
SlideDame		C APP BouncingBalls_FPU.c	2017/12/05 15:28	C source file
SideDemo		C APP BouncingBalls NOFPU.c	2017/12/05 15:28	C source file
Conng		H APP BouncingBalls Private.h	2017/12/13 15:53	C header file
⊳ 🌆 drw_2d		C APP_DrawingEngine.c	2017/12/13 17:43	C source file
🛛 👑 emWin_RX65N		C APP_HelpDialog.c	2017/10/20 14:40	C source file
🐌 GUI		C APP_Mandelbrot.c	2019/07/19 11:01	C source file
> 퉲 r_bsp		H APP_Mandelbrot_DrawMandelbrot.h	2017/10/20 14:41	C header file
🛛 📗 r_cmt_rx		C APP_Mandelbrot_FPU.c	2017/10/20 14:41	C source file
🐌 r_config		C APP_Mandelbrot_NOFPU.c	2017/10/20 14:41	C source file
Image: Participation of the second		H APP_Mandelbrot_Private.h	2017/12/13 17:39	C header file
🛯 🛄 r flash rx		C APP_StartScreen.c	2017/10/20 14:41	C source file
		C APP_Update.c	2018/01/09 17:34	C source file
		H] EnvisionKit.h	2017/12/13 17:41	C header file
	_	C] MainTask_EnvisionKit.c	2019/07/29 14:36	C source file
⊳ <mark>∥₀</mark> r_irq_rx	=	H Resource.h	2017/12/13 16:23	C header file
r_pincfg				
⊳ 퉲 r_sci_iic_rx				
Image: Participation of the second				
Image: big is a standard in the standard in				
🛛 📗 r_tfat_rx				
> 퉲 r_usb_basic				
⊳ 퉲 r_usb_hmsc				
🌗 src				
🐌 Tool				

< emWin_RX65N >

 Application EnvisionKit MT_ScratchPad SlideDemo	・・・・・・ Envision キット用サンプルアプリケーション ・・・・・ マルチタッチ機能のデモアプリケーション ・・・・・ マルチウインドウスライドデモアプリケーション
⊢ Config	GUI モジュール・LCD ドライバコンフィグレーション設定
⊢ GUI	GUI モジュールライブラリ及びヘッダ



• RX65N Envisionボードについて



マイコンデバイス:	RX65N(ROM:2MB、内蔵 RAM:640KB)
LCD パネル:	TFT WQVGA (480 x 272)
デバッガ I/F:	オンボード E2-Lite デバッガ(外部デバッガサポート有)

※RX65N Envision キットの詳しい仕様については取扱説明書をご参照ください。

● RX65N Envision ボード設定

デバッグモードの)設定:	SW1-	-1:ON	SW1-2:0FF	SW4-1:OFF
NO	3₩4-1		OFF	プログラマ・デバッ	ッガモード
	SM/1_1		ON	USB シリアルモート	n
NO	3W1-2		OFF	CPU リセット無効に	こ設定します
S at	SW1 2		ON	CPU リセット有効	(LOW)に設定します
	3W1-1	_	OFF	スタンドアロンモ-	$- \models$
21	CW1 1		ON	プログラマ・デバッ	ッガモード



3.1. デモサンプル

3.1.1. 標準サンプルアプリケーション

デモアプリケーションは以下の4種類のサンプルが含まれています。

2D Drawing Engine: グラフィックアクセラレータを有効にしたサンプル

Bouncing balls:

バウンスボールサンプル (Dave2D エンジンを使用します)

Mandelbrot calculation: マンデルブロ集合 (単精度浮動小数点ユニットを使用します)

Firmware Bank Swap Demo:

ユーザファームウェアの更新を行います





3.1.2. マルチタッチ機能のデモアプリケーション

emWin のマルチタッチ機能の使用方法を示すデモサンプル:



3.1.3. マルチウインドウスライドデモアプリケーション



● 開発環境

コンパイラ: Renesas RXCC コンパイラ (v3.00.00 以降)
 IDE: Renesas e²Studio (Version 2022-07 以降)
 CS+ for CC (V8 以降)



3.2. プロジェクトビルド方法 (e2Studio)

• Step1:e²Studio IDE を起動してプロジェクトを開きます

[ファイル]→ [インポート]コマンドで emWin_RX65N フォルダ下のプロジェクトをワークスペースにインポート します。





i workspace - e ² studio		_		×
ファイル(E) 編集(E) ソース(S) リファクタリング(T) ナビゲート(N) 検索(A) プロジェクト(<u>P</u>) Renesas <u>V</u> iews 実行(<u>R</u>) ウィンドウ(<u>W</u>) ヘルプ(<u>H</u>)			
🔦 🏘 🔳 🏘 デバッグ(B) 🗸 🖻 RX65N_SlideDemo Har 🗸	│ _ →	• 🚳 • 🖻 • 🎯	- 10	<i>?</i> •
◎圖 # 過 ▼ 前 ▼ や や や ▼ ♂	Q (昭 昭 C/C++ #	スマート・コンフィク	レータ *	* デバッグ
▶ プロジェクト・エクスプローラー ×		調 アウトライン ×		- 0
✓		アウトラインを提供す	ロスアクテ	ィブなエデ
> 🔊 Includes		ィターはありません。		
> 🐸 Application				
> 😂 Config				
> 🐸 GUI				
> 🙆 drw_2d				
> 😂 r_bsp				
> 🐸 r_cmt_rx				
> 🐸 r_config				
> @ r_dmaca_rx				
> @ r_flash_rx				
> @ r_glcdc_rx				
> @ r_gpio_rx	□□ 同題 □ コンソール × (♥ スマート・ノフワザー い スマート・マニュアル		w . L	
> @ r_irq_rx	現在、表示するコンソールかめりません。			
> constanti lis su				
> @ r_sys_ume_rx				
> @ r tfat ry				
> @ rush hasic				
> er usb hmsc				
> esc				
makefile.init				
RX65N EnvisionKit HardwareDebug Double64Bit.launch				
RX65N_EnvisionKit HardwareDebug.launch				
RX65N_MT_ScratchPad HardwareDebug.launch				
RX65N_SlideDemo HardwareDebug.launch				
🖹 userprog.rsu				
< >				
	⊯ emWin_RX65N			
コンテキスト・ヘルプをポーリング中: (55%)				

● Step2:プロジェクトビルド設定内容を確認します

プロジェクトプロパティダイアログを開いて、「C/C++ビルド → 設定」タブ下のビルド環境設定内容を確認しま す。インストールされている CC-RX ツールチェーンバージョンとサンプルプロジェクトの設定が異なる場合は、 CC-RX ツールチェーンバージョンを正しく設定します。

フィルタ入力	設定	
> リソース ~ C/C++ ビルド スタック解析 ツールチェイン・エ	Configuration: HardwareDebug (アクティブ)	Enable toolchain integration 現在のツールチェーン
ビルド変数 ロギング 環境 設定	◎ ツール設定 Tookhain Device ドビルド・ステップ デビルド成果物 ◎ パイナリー・パーサー ● I ご Enable tookhain integration 現在のツールチェーン	ッールチェーン: Renesas CCRX パージョン: v2.07.00
》C/C++ 一般 Renesas QE	ツールチェーン: [®] Renesas CCRX パージョン: v2.07.00	ツールチェーンの変更 (タブを切り替える前に「適用」ボタンをクリックしてください)
Task Tags	ツールチェーンの変更 (タブを切り替える前に「適用」ボタンをクリックしてください)	ツールチェーン: Renesas CCRX ~
Validation ビルダー プロジェクト・ネーチ: プロジェクト参照 実行/デパッグ設定	ツールチェーン: ~ パージョン: ~	パージョン: v3.04.00 ~



Step3:プロジェクト(デバッグ構成)をビルドします

workspace - e² studio

> 🥴 Config

> 🥴 drw 2d

> > > > Pr_cmt_rx

> 🥴 r dmaca_rx

> 😕 r_flash_rx

> 🤒 r_bsp

> 🥴 GUI

ファイル(F) 編集(E) ソース(S) リファクタリング(T) ナビゲート(N) 検索(A) プロジェクト(P) Renesas Views 実行(R) ウィンドウ(W) ヘルプ(H) 🔏 🏘 🔳 🏘 デバッグ(B) RX65N_SlideDemo Har * プロジェクトを開く(E) | ☆ ▼ 94 ▼ 105 ▼ 花 110 ▼ 000 16 プロジェクトを閉じる(S) 월 • 월 • 한라라 • • • 년 **Open FSP Configuration** E \$ 7 8 - C ┗ プロジェクト・エクスプローラー ■ すべてビルド(A) Ctrl+Alt+B ビルドはな > S Includes > 🥴 Application プロジェクトのビルド(B) Ctrl+B

ワーキンク・セットのビルド(W)

>

5

Alt+D

クリーン(N)...

自動的にビルド(M)

C/C++ インデックス(i)

■ すべての依存関係を更新

ビルド・ターゲット

Change Device



ビルド後に「HardwareDebug」フォルダ下に「RX65N_EnvisionKit.x」デバッグ式 ELF イメージが作成されます。



3.3. ダウンロード・デバッグ方法

Renesas e²Studio IDE 環境でのフラッシュダウンロード・デバッグ操作にデバッグ構成が必要です。 RX65N Envision キット用サンプルに「RX65N_EnvisionKit HardwareDebug」デバッグ構成が含まれています。 以下の 2 種類のデバッグ設定方法を解説します。

③ 「RX65N_EnvisionKit HardwareDebug」デバッグ構成を使用してデバッグ

④ 新規デバッグ構成を作成して行うデバッグ操作

● Step1:ボードをデバッグモードに設定します

ボード上の DIP-SW を以下のように設定します。

SW1-1:ON SW1-2:OFF SW4-1:OFF

キット付属ケーブルでボード上のmini-USB コネクタ(CN9)をパソコンに接続します。

● Step2:「RX65N_EnvisionKit HardwareDebug」デバッグ構成を確認します

💽 workspace - e² studio

ファイル(F) 編集(E) ソース(S) リファクタリング(T) ナビ	【ゲート(N) 検索(A) フ	ロジェクト(P) Ren	nesas Views	実行(R) ウィンドウ(W) ヘルプ(H
🔏 🐲 🔳 🎋 デバッグ(B) 🗸 🖾 RX65N_Env	risionKit Har 🔌 👘	9 - B G B -	- 🔦 - 📓 📼	1 🐐 × & @ 🇞 🏘 🕶 🍕	
월 · 원 · 우리수 · 우 · [년					
▲ プロジェクト・エクスプローラー×	nvisionKit HardwareD	Debug	7 編集	アイコンを 🔪	
✓ SemWin RX65N [HardwareDel States No. 1]	ideDemo Hardware	Debug		フリック ノー	
> W パイナリー E RX65N M	T ScratchPad Hardw	areDebug			
- Minder	-joraten da Haran	dicbebug			
				_	
(2) 構成の編集		↓	×		
Edit Renesas GDB Hardware Debugging configuration I	RX65N_EnvisionKit Har	dwareDebug for	デバッグ(D)		
				1	
Launch Configuration Name: RX65N_EnvisionKit HardwareDebug					
■ X1 / * Debugger ► Startup □ 共通(C) * ソース					
70919F(P):			40 077 /B1	1	
emwin_KXooN			奓照(B)	1	
HardwareDebug¥RX65N EnvisionKit x				1	
International Content	変数(V) ブ	ロジェクトの検索(H)	参照(R)		
起動前に必要に応じてビルド					
Build Configuration: HardwareDebug			~		
○ 自動ビルドを有効にする	○ 自動ビルドを無効にする				
● ワークスペース設定の使用	ワークスペース設定の構成				
⑦ 複製 削除		OK	キャンセル		



 \times

構成の編集

Edit Renesas GDB Hardware Debugging configuration RX65N_EnvisionKit HardwareDebug for デバッグ(D)

メイン 🌣 Debugger 🕨 Startup 💷 共通(C) 💱 ソース		
Debug hardware: E2 Lite (RX) \checkmark Target Device: R5	F565NE_DI	
GDB Settings Connection Settings デバッグ・ツール設定		
✓ クロック		
メイン・クロック・ソース	EXTAL	~
EXTAL 周波数[MHz]	27.0	
動作周波数 [MHz]		
内部フラッシュメモリー書き換え時にクロック・ソースの変更を	許可する はい	~
✔ ターゲット・ボードとの接続		
エミュレーター	(Auto)	
接続タイプ	JTag	~
JTag クロック周波数[MHz]	6.00	~
Fine ボーレート[Mbps]	1.50	~
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	~
✓ CPU 動作モード		
レジスター設定	シングルチップ	~
モード端子	シングルチップ・モード	~
起動バンクを変更する	はい	~
起動バンク	パンク1	~
◇ 通信モード		
€−ド	デバッグ・モード	~
デバッガー終了後にユーザー・プログラムを実行する	いいえ	~
✓ フラッシュ		
ID J-K	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	

● Step3:デバッグセッションを開始します

🔯 workspace - e² studio					
ファイル(F) 編集(E) ソース(S) リファクタリン	<u>パ(円) ナビガート(N) 絵索(A) プ</u> ロジェクト(P) Renesas Vie	ws 実行	〒(R) ウィンドウ(W) ヘルプ(H)		
🤏 🔯 🔳 🏶 デバッグ(B) 🕐 🖾 R2	X65N_EnvisionKit Har 🗸 🏟 🚦 🕄 🖛 🗟 🕲 🖛 🗞 🕶	B	Renesas Debug Tools	>	2 0 9
1 · · · · · · · · · · · · · · · · · · ·		0.	宝行(R)	Ctrl+F11	
┗ プロジェクト・エクスプローラー ×	E 🕸 🎖 💈 🗖 🖬	16	デバッグ(D)	F11	
w SemWin_RX65N [HardwareDebug]	1		実行履歴(T)	>	
> 💐 バイナリー		0	実行(S)	>	
> 🔊 Includes			実行構成(N)		
Application			デバッグ履歴(H)	>	1
> 🤒 Config		*	デバッグ(G)	>	
> 🐸 GUI			デバッグの構成(B)		
b Cuah Ba			e e e e e e e e e e e e e e e e e e e		1





● Step4:ボードを RUN モードに設定します

デバッグ操作が不要の場合は、ボード上の DIP-SW を以下のように変更してスタンドアロンモードを設定します。
 SW1-1:0FF SW1-2:0FF SW4-1:0FF



3.4. CS+ IDE 環境への移植方法

3. 4. 1. e2Studio プロジェクトを CS+ IDE にインポート

● Step1:e²Studio IDE を起動してプロジェクトを開きます

[ファイル]→ [インポート]コマンドで emWin_RX65N フォルダ下のプロジェクトをワークスペースにインポート

します。

THE OTHER DESIGNATION OF THE OTHER OF THE OTHE	Alt+S/76+NA	インポート	- 0	×
ファイルを開く()				~
ファイル・システムからプロジ:	ェクトを開く	選択		PKg
最近のファイル	>	アーカイブ・ファイルまたはディレクトリーから新規プロジェクトを作成します。		
閉じる(C)	Ctrl+W			
すべて閉じる(L)	Ctrl+シフト+W	インポート・ウィザードの理択(5):		
保存(S)	Ctrl+S			
 名前を付け(床仔(A) すべて保存(E) 	Ctrl+シフト+S			
前回保存した状態に戻す(T)	□ ファイル・システム		<u> </u>
移動(V)		- ノノリン		
名前を変更(M)	F2	☆ 既存プロジェクトをワークスペースへ		
 1 更新(F) 行区切り文字の変換の) 	F5	10. 放定		
11区 99 文子の支援(0)		> 🗁 C/C++		
ニ インボート(1)		> 🗁 Oomph		~
ロ エクスホート(O)				
プロパティ(R)	Alt+Enter	↓ ↓		
ワークスペースの切り替え(W	/) >			
再開		⑦ < 厚る(B) 次へ(N) > 終了(F)	キャンナ	11
終了(X)		C CONTRACT CONTRACTOR		
プロジェクトをインポート		- C X		
インボート プロジェクトをインボート 既存の Eclipse プロジェクトを検索	さするディレクトリーを選択します。			
 インポート プロジェクトをインボート 既存の Eclipse プロジェクトを検索 ルート・ディレクトリーの選択(I): [にするディレクトリーを選択します。 C.¥Workspace¥RX¥emWin-RX-220603¥em	- □ × ////////////////////////////////////		
 インボート プロジェクトをインボート 既存の Eclipse プロジェクトを検索 ルート・ディレクトリーの選択(I): [アーカイブ・ファイルの選択(A): [ミするディレクトリーを選択します。 C:¥Workspace¥RX¥emWin-RX-220603¥em	- □ × ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓		
・ インボート ・ プロジェクトをインボート 既存の Eclipse プロジェクトを検索 ・ ルート・ディレクトリーの選択(口): ・ アーカイブ・ファイルの選択(A): プロジェクト(P): ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	なるディレクトリーを選択します。 C:¥Workspace¥RX¥emWin-RX-220603¥em	- □ × Win_RX65N ✓ 参照(R) 参照(R)		
インボート ガロジェクトをインボート 既存の Eclipse ブロジェクトを検索 ルート・ディレクトリーの選択(ロ): [アーカイブ・ファイルの選択(A): [ブロジェクト(P): ✓ emWin_RX65N(C:¥Work)	?するディレクトリーを選択します。 C:¥Workspace¥RX¥emWin-RX-220603¥em space¥RX¥emWin-RX-220603¥emWin_RX6	- □ × デ Win_RX65N ✓ 参照(R) 参照(R) 35N) すべて選択(S) 思わたたたいて変形のの		
 インボート プロジェクトをインボート 既存の Eclipse プロジェクトを検索 ルート・ディレクトリーの選択(①: アーカイブ・ファイルの選択(A): プロジェクト(P): emWin_RX65N(C:¥Work) 	Rするディレクトリーを選択します。 C:¥Workspace¥RX¥emWin-RX-220603¥em space¥RX¥emWin-RX-220603¥emWin_RX6	- □ × デ Win_RX65N ✓ 参照(R) 参照(R) 参照(R) 参照(R) 変照(R) 変照(R) 変照(R) 変照(R)		
 Aンボート Diラェクトをインボート 既存の Eclipse プロジェクトを検索 ルート・ディレクトリーの選択(1): [) アーカイブ・ファイルの選択(A): [プロジェクト(P):	ミするディレクトリーを選択します。 C:¥Workspace¥RX¥emWin-RX-220603¥em space¥RX¥emWin-RX-220603¥emWin_RX6	- □ × win_RX65N ◆ 参照(<u>B</u>) 参照(<u>B</u>) 参照(<u>B</u>) すべて選択(<u>S</u>) 選択をすべて解除(<u>D</u>) 更新(<u>E</u>)		
 3 インボート プロジェクトをインボート 既存の Eclipse プロジェクトを検索 シルート・ディレクトリーの選択(口): [ファーカイブ・ファイルの選択(ム): [プロジェクト(P): ✓ emWin_RX65N(C:¥Work オブション スストレ,たプロジェクトを検索に 	まするディレクトリーを選択します。 C.¥Workspace¥RX¥emWin-RX-220603¥em space¥RX¥emWin-RX-220603¥emWin_RX6	- □ × Win_RX65N ✓ 参照(£) 参照(£)		
 ▲ 1ンボート プロジェクトをインボート 既存の Eclipse ブロジェクトを検索 ● ルート・ディレクトリーの選択(①: [○ アーカイブ・ファイルの選択(Δ): [プロジェクト(型): ✓ emWin_RX65N(C:¥Work オブション ✓ ネストしたプロジェクトを検索(Ⅱ □ プロジェクトをワークスペースにコ 	?するディレクトリーを選択します。 C:¥Workspace¥RX¥emWin-RX-220603¥em space¥RX¥emWin-RX-220603¥emWin_RX6) ビー(C)	- □ × Win_RX65N ✓ 参照(R) 参照(R) 55N) すべて選択(S) 選択をすべて解除(D) 更新(E)		
 ▲ 1ンボート プロジェクトをインボート 既存の Eclipse ブロジェクトを検索 ● ルート・ディレクトリーの選択(ロ): □ アーカイブ・ファイルの選択(ム): ゴロジェクト(P): ☑ emWin_RX65N(C:¥Work) オブション ☑ ネストしたプロジェクトを検索(田 □ プロジェクトをワークスペースにコ □ 完」 次策,新しくインボートした 	?するディレクトリーを選択します。 C:¥Workspace¥RX¥emWin-RX-220603¥em space¥RX¥emWin-RX-220603¥emWin_RX6) ビー(C) プロジェクトを閉じる(g)	- □ × Win_RX65N ✓ 参照(B) 参照(B) 参照(C) すべて選択(S) 選択をすべて解除(D) 更新(E)		
 ↓ インボート ⑦ ロジェクトをインボート 既存の Eclipse プロジェクトを検索 ⑦ ルート・ディレクトリーの選択(D): [⑦ アーカイブ・ファイルの選択(A): [⑦ アーカイブ・ファイルの選択(A): [⑦ アーカイブ・ファイルの選択(A): [⑦ マーカイブ・ファイルの選択(A): [⑦ マーガイブ・ファイルの選択(A): [⑦ マーガイブ・ファイルの選択(A): [⑦ マーガスペースに既に存在するブ ◎ マークスペースに既に存在するブ ◎ マークスペースに既に存在するブ 	Rするディレクトリーを選択します。 C:¥Workspace¥RX¥emWin-RX-220603¥em space¥RX¥emWin-RX-220603¥emWin_RX6) ビー(C) プロジェクトを開じる(<u>o</u>) 「ロジェクトを隠す(j)	- □ × 「「「RX65N ◇ 参照(<u>R</u>)… 参照(<u>R</u>)… 参照(<u>R</u>)… 35N) すべて選択(<u>S</u>) 選択をすべて解除(<u>D</u>) 更新(<u>E</u>)		
 ▲ 1ンボート プロジェクトをインボート 既存の Eclipse ブロジェクトを検索 ルート・ディレクトリーの選択(口): [アーカイブ・ファイルの選択(Δ): プロジェクト(P): ✓ emWin_RX65N(C:¥Work オブション ダ ネストしたプロジェクトを検索(Ц □ プロジェクトを検索(Ц □ プロジェクトを検索(L) □ デン茨系、新しくインボートした □ ワークスペースに既に存在するブ ワーキング・セット □ ワーキング・セット 	まするディレクトリーを選択します。 C:¥Workspace¥RX¥emWin-RX-220603¥em space¥RX¥emWin-RX-220603¥emWin_RX6) ビー(C) ジロジェクトを閉じる(g) fロジェクトを隠す(j) を追加rCh	- □ × Win_RX65N ✓ 参照(£) 多照(£)		
 ▲ 1ンボート プロジェクトをインボート 既存の Eclipse ブロジェクトを検索 ● ルート・ディレクトリーの選択(ロ): [⑦ アーカイブ・ファイルの選択(Δ): [⑦ アーカイブ・ファイルの選択(Δ): [⑦ ロジェクト(P): ⑦ emWin_RX65N(C:¥Work) オブション ※ イストしたプロジェクトを検索(田 ⑦ ブロジェクトを欠つクスペースにコ ○ デンダ、新しくインボートした ⑦ ワークスペースに既に存在するブ ⑦ ーキング・セット(こプロジェクト) ⑦ ワーキング・セット(こプロジェクト) 	<pre>?するディレクトリーを選択します。 C:¥Workspace¥RX¥emWin-RX-220603¥em space¥RX¥emWin-RX-220603¥emWin_RX6) ビー(C) :プロジェクトを開じる(<u>o</u>) /ロジェクトを聴す(j) を追加(<u>T</u>)</pre>	- □ × Win_RX65N × 参照(B) 参照(B) 参照(B) すべて選択(S) 選択をすべて解除(D) 更新(E)		
 ▲ 1ンボート プロジェクトをインボート 既存の Eclipse ブロジェクトを検索 ● ルート・ディレクトリーの選択(ロ): ○ アーカイブ・ファイルの選択(Δ): 「ロジェクト(P): ✓ emWin_RX65N(C:¥Work オブション ダ ネストしたプロジェクトを検索(H □ プロジェクトをワークスペースにコ □ 完了次第、新しくインボートした □ ワークスペースに既に存在するブ ワーキング・セット(Ω): 	 ?するディレクトリーを選択します。 C:¥Workspace¥RX¥emWin-RX-220603¥em space¥RX¥emWin-RX-220603¥emWin_RX6)) ビー(C) プロジェクトを閉じる(<u>o</u>) ?ロジェクトを隠す(j) を追加(T) 	- □ × Win_RX65N ◇ 参照(B) 参照(B) ずべて選択(S) 選択をすべて解除(D) 更新(E)		



workspace - e* studio		-		×
ファイル(E) 編集(E) ソース(S) リファクタリング(T) ナビゲート(N) 検索(A	A) プロジェクト(P) Renesas <u>V</u> iews 実行(R) ウィンドウ(W) ヘルプ(H)			
		• 🚳 • 🔂 • 🞯	- 10 1	P 👻
※回 # 包 ▼ 初 ▼ や や や ▼ ピ	Q. IB ঊC/C++	₽ スマート・コンフィグ	L-9 #	デバッグ
<u>▶ プロジェクト・エクスプローラー ×</u> 🗈 🕏 🖓 🖁 🗖 🖬	- 0	器 アウトライン ×		
✓ emWin RX65N [HardwareDebug]		アウトラインを提供す	るアクティ	イブなエデ
> Includes		イターはありません。	0///	12.911
> 😕 Application				
> 🐸 Config				
> 🐸 GUI				
> 🐸 drw_2d				
> 🤒 r_bsp				
> 🤒 r_cmt_rx				
> 😕 r_config				
> 🐸 r_dmaca_rx				
> 🥴 r_flash_rx				
> 🐸 r_glcdc_rx				
> @ r_gpio_rx	型 問題 □ コンソール × ● スマート・フラウザー □ スマート・マニュアル		2 T 13	• - 0
> @ r_irq_rx	現在、表示するコンソールがありません。			
>				
> 🐸 r_sci_lic_rx				
> @ r_sys_time_rx				
> @ r_tfat_orver_tx				
> @ rush basic				
> @ r usb hmsc				
> esc				
makefile.init				
RX65N_EnvisionKit HardwareDebug_Double64Bit.launch				
RX65N_EnvisionKit HardwareDebug.launch				
RX65N_MT_ScratchPad HardwareDebug.launch				
RX65N_SlideDemo HardwareDebug.launch				
🖹 userprog.rsu				
< >				
	🥩 emWin_RX65N			
コンテキスト・ヘルプをポーリング中: (55%)				

● Step2:プロジェクトを RCPC ファイルとしてエクスポートします

[ファイル]→ [エクスポート]コマンドで emWin_RX65N プロジェクトを CS+ 以降と互換性がある RCPC 式ファイ ルにエクスポートします。





「一般 → Renesas 共通プロジェクト・ファイル」を選択します。

📴 エクスポート			<u>88</u>		×	
選択					R79	
CS+のインポート用にルネサン	ス共通プロジェクト・ファイ	イル形式でプロジェクトを	エクスポートします。			
エクスポート・ウィザードの選択	₹(S) :					
フィルタ入力						
~ ≫一般					^	
Renesas FSP Arc	er Pack	-				
🖙 Renesas共通プロ	ジェクト・ファイル					
<u>□ ァ_カイブ・ファイル</u>					~	
0	< 戻る(B)	次へ(N) >	終了(F)	キャンセ	zJV	
國 プロジェクトのエクスポ	²−ト					×
 プロジェクトのエクスボ 共通プロジェクト・ファイ 	ト (ルにエクスポート					×
プロジェクトのエクスポ 共通プロジェクト・ファイ 選択した e2 studio プロ す。	ピート イルにエクスポート Iジェクトを、共通プロ・	ジェクト・ファイル・フォ	-マット V1.0 にエク	ー		×
 プロジェクトのエクスポ 共通プロジェクト・ファイ 選択した e2 studio プロ す。 選択されたプロジェクト: 	ピート イルにエクスポート リジェクトを、共通プロ:	ジェクト・ファイル・フォ	-マット V1.0 にエク	ー.	L E	×
プロジェクトのエクスポ 共通プロジェクト・ファイ 選択した e2 studio プロ す。 選択されたプロジェクト: 共通プロジェクト・ファイルの	ピート イルにエクスポート リジェクトを、共通プロ・ のデフォルトのロケーシ:	ジェクト・ファイル・フォ ョンは、選択したプロジ	-マット V1.0 にエク デェクトのルート・ディ	ー マスポートしま レクトリーで	E.	×
プロジェクトのエクスボ 共通プロジェクト・ファイ 選択した e2 studio プロ す。 選択されたプロジェクト: 共通プロジェクト・ファイルの CS+にインポートする時は、	パート イルにエクスポート ロジェクトを、共通プロ・ のデフォルトのロケージ: 先に emWin_RX65	ジェクト・ファイル・フォ ョンは、選択したプロジ N.rcpc を更新してく3	ーマット V1.0 にエク 「ェクトのルート・ディ ださい。	レクトリーで	E t	×
 プロジェクトのエクスポ 共通プロジェクト・ファイ 選択した e2 studio プロ す。 選択されたプロジェクト: 共通プロジェクト・ファイルの CS+にインポートする時は、 ロケーション: た:/Worksp 	アート イルにエクスポート ロジェクトを、共通プロ・ のデフォルトのロケーショ 先に emWin_RX65 bace/RX/emWin-RX	ジェクト・ファイル・フォー ヨンは、選択したプロジ N.rcpc を更新してく (-220603/emWin_f	ーマット V1.0 にエク 「ェクトのルート・ディ ださい。 XX65N/emWin_R	マスポートしま レクトリーで X65N.rcp0	 بخ.	× !参照!
プロジェクトのエクスポ 共通プロジェクト・ファイ 選択した e2 studio プロ す。 選択されたプロジェクト: 共通プロジェクト・ファイル CS+にインポートする時は、 ロケーション: C:/Worksp	ピート イルにエクスポート ロジェクトを、共通プロ・ のデフォルトのロケーシ: 先に emWin_RX65 pace/RX/emWin-RX	ジェクト・ファイル・フォ ヨンは、選択したプロジ N.rcpc を更新してく (-220603/emWin_f	-マット V1.0 にエク デェクトのルート・ディ ださい。 XX65N/emWin_R	ー スポートしま レクトリーで X65N.rcp(E Total	× 参照
 ブロジェクトのエクスボ 共通プロジェクト・ファイ 選択した e2 studio プロ す。 選択されたプロジェクト: 共通プロジェクト・ファイル CS+にインポートする時は、 ロケーション: C:/Worksp 	ピート イルにエクスポート Dジェクトを、共通プロ・ のデフォルトのロケーシ: 先に emWin_RX65 pace/RX/emWin-RX	ジェクト・ファイル・フォ ョンは、選択したプロジ N.rcpc を更新してく (-220603/emWin_f	-マット V1.0 にエク デェクトのルート・ディ ださい。 XX65N/emWin_R	ー スポートしま レクトリーで X65N.rcpt	E Tato	× !参照!
 プロジェクトのエクスボ 共通プロジェクト・ファイ 選択した e2 studio プロ す。 選択されたプロジェクト: 共通プロジェクト・ファイルの CS+にインボートする時は、 ロケーション: た:/Worksp 	ピート イルにエクスポート Iジェクトを、共通プロ・ Dデフォルトのロケーショ 先に emWin_RX65 bace/RX/emWin-RX	ジェクト・ファイル・フォ ョンは、選択したプロジ N.rcpc を更新してく (-220603/emWin_f	-マット V1.0 にエク ^デ ェクトのルート・ディ ださい。 XX65N/emWin_R	ー スポートしま レクトリーで X65N.rcpd	ि हे द	× 参照

● Step3:プロジェクトを閉じて e2Studio を終了します

📴 workspace - e² studio

ファイル(F) 編集(E) ソース(S) リファクタリング(T) ナビゲート(N) 検索(A) プロジェクト(P) Renesas Views 実行(R) ウィンドウ(W) ヘルプ(H)

9	💺 🔳 🏘 デバッグ(B) 🗸 🗷	🛚 RX65N_SlideDemo Har 🗸 🐐	_	プロジェクトを開く(に)	-	🎋 🕶	Q. ▼ 0 ₀ ▼ 株	5 DIN 1
2				プロジェクトを閉じる(S)				
•	プロジェクト・エクスプローラー ×	🖻 😫 🍞 💈 🗖 C		Open FSP Configuration	プロジェクトた明パスの	2.53		
~	📁 emWin_RX65N [HardwareDeb	oug]	220	すべてビルド(A)	Ctil+Alt+B	(3)		
	> 歳 パイナリー			ビルド構成	>			
	> 🔊 Includes			プロジェクトのビルド(B)	Ctrl+B			
	> 🥝 Application			1-+				



● Step4:プロジェクトを CS+環境で開きます

開発環境: CS+ for CC V8.08.00

CS+ IDE を起動して、[プロジェクト] \rightarrow [プロジェクトを聞く..]コマンドを事項し、Step2 で作成されました e2Studio エクスポートプロジェクトファイル「emWin_RX65N. rcpc」を開きます。

🔘 CS+ for CC - [スタート]							
ファイル(F) 編集(E) 表示(V) プロ	リジェクト	(P) ビルド(B) デバッグ(D) ツール(T) さ	レインドウ(W) ヘルプ(H)		
🍇 スタート(S) 🔒 📑 🚮	新しいフ	プロジェクトを作成(N)		D0%	• 67 67		- 🕺
I ((() () () () () () () () (プロジュ	:クトを開く(P)					
プロジェクト・ツリー 中 >	お気に2	、りのプロジェクト(F)		•			
2 🕜 🙎 😂	追加(D)			•			
🕼 プロジェクトを開く						×	
\leftarrow \rightarrow \checkmark \uparrow \blacksquare \ll RX > emWin-RX	(-220603	emWin_RX65N	ٽ ~	,	_RX65Nの検索		
整理 ▼ 新しいフォルダー					III 🔹	?	
emWin-RX-220603	^	名前		サイズ	更新日時	^	
emWin_RX65N		r_sci_iic_rx			2022/09/23 18:17		
emWin_RX72N		r_sys_time_rx			2022/09/23 18:17		
settings		r_tfat_driver_rx			2022/09/23 18:17		
Application		r_tfat_rx			2022/09/23 18:17		
		r_usb_basic			2022/09/23 18:17		
Contig		r_usb_hmsc			2022/09/23 18:17		
GUI		src			2022/09/23 18:17		
HardwareDebug		emWin_RX65N.rcpc		159 KB	2022/09/24 12:21	~	
src	×	<				>	
ファイル名(N) emWir	n_RX65N.r	срс	~	MCU Simula	tor Online∕e² stud	\sim	
				MCU Simulat	アイル (************************************	o用プロジェク ccnii)	^ッ ト・ファイル (*.rcpc)
				HEW用ワーク HEW用プロジ PM+用ワーク PM+用プロジ	スペース・ファイル (*.hw ェクト・ファイル (*.hwp) スペース・ファイル (*.prv ェクト・ファイル (*.prj)	s) v)	



● Step5:プロジェクトマイコンデバイス設定を変更します

プロジェクト変換設定			×	7			
プロジェクト(P):							
② 説明 【TemWin_RX65N	プロジェクト設定 変換先マイクロコントローラ マイクロコントローラ(T):	RX					
	使用するマイクロコントロー	使用するマイクロコントローラ(M)					
	🏭 (マイクロコントローラを枯	食索できます) アップデート(U)					
	R5F565NED×EG	3-DUAL(176pin へ 品種名:R5F5657 (144pin) 内蔵ROMサイズ - DUAL(144pin) 追門情報:Packa	IEDxFB_DUAL K/ቫイト]:1024 /ቫイト]:655360 ge=PLQP0144KA-B				
	R5F565NEDxFC R5F565NEDxFC R5F565NEDxFC	2(176pin) 2-DUAL(176pin) 2(100pin)	\sim	品種名: R5F565NEDxFB_DUAL	\bigcirc		
	変換先プロジェクト プロジェクトの種類(K):	空のアプリケーション(CC-RX)	×				
	プロジェクト名(N):	emWin_RX65N					
	□ 変換元ブロジェクトと異な	3る場所に作成する(A)					
	作成場所(L):	C:¥Users¥santo¥OneDrive¥ドキ	ュメント¥CS+ Pr 📦 🗸 参照(R)				
	☑ 変換元ブロジェクト・	✓ 変換元ブロジェクト・フォルダのすべてのファイルを作成場所にコピーする(0)					
	☑ 変換直後のプロジェクト権	専成ファイルをまとめてバックアップする(C)					
L		ОК	キャンセル ヘルプ(H)]			

(③ emWin_RX65N - CS+ for CC - [ブロパティ] - □	×
ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D) ツール(T) ウインドウ(W) ヘルプ(H)	🧭 🙆
🚳 スタート(S) 🔒 🔒 🕼 🖄 🐚 🖄 🖛 🔍 🔹 👘 🐨 🔹 👘 🐨 🔹 👘	
- 🖓 🖓 🐙 🌠 : 🔲 💭 📮 : 🌈 VJューション一覧(S)	
😳 לייער איז 🔁 איז	- x
Config Gul Gul	- +
Control (Control (Contro) (Control (Contro) (Control (Contro) (Contro) (Contro) (Contr	
Fi 20/57 / パー F2 名前の変更 F3 次を検索 F1 次を置換 F5 実行 F5 ビルド8 デパー F7 ビルド・ブロー F0 ブレークせず… F9 ブレークせず… F1 ステップ・オー・・・ F1 ステップ・イン F12 開設また A 非接続	Eは変…



3.4.2. プロジェクト設定を修正

● Step6:コンパイル・オプション設定を修正します



コピーして新しいパスを設定します。

%ProjectDir%¥r_glcdc_rx %ProjectDir%¥r_gpio_rx %ProjectDir%¥r_sys_time_rx %ProjectDir%¥r_usb_hmsc¥src¥inc %ProjectDir%¥r_usb_basic¥src¥driver¥inc %ProjectDir%¥r_usb_basic¥src¥hw¥inc %ProjectDir%¥r_tfat_rx¥lib %ProjectDir%¥r_tfat_driver_rx¥src %ProjectDir%¥r_flash_rx¥src %ProjectDir%¥r_flash_rx¥src¥flash_type_4 %ProjectDir%¥r_flash_rx %ProjectDir%¥r_tfat_driver_rx %ProjectDir%¥drw_2d¥inc¥tes %ProjectDir%¥r_tfat_rx %ProjectDir%¥r_usb_basic %ProjectDir%¥r_usb_hmsc %ProjectDir%¥r_sci_iic_rx¥src %ProjectDir%¥r_sci_iic_rx %ProjectDir%¥r_irq_rx %ProjectDir%¥r_cmt_rx %ProjectDir%¥r_bsp¥mcu¥rx65n %ProjectDir%¥r_bsp¥mcu¥rx65n¥register_access %ProjectDir%¥GUI %ProjectDir%¥Config %ProjectDir%¥r_bsp %ProjectDir%¥r_bsp¥board¥generic_rx65n %ProjectDir%¥r_config %ProjectDir%¥r_pincfg %ProjectDir%¥r_dmaca_rx







サンプルプロジェクトの以下のファイルは個別コンパイル・オプション設定が必要ですので、同様に新しいパス を設定します。

- emWin_RX65N¥Application¥EnvisionKit¥APP_BouncingBalls_NOFPU.c
- emWin_RX65N¥Application¥EnvisionKit¥APP_Mandelbrot_NOFPU.c



Step7:リンク・オプション設定を修正します

🙆 emWin_RX65N - CS+ for CC - [プロパティ]		
ファイル(<u>E</u>) 編集(<u>E</u>) 表示(<u>V</u>) プロジェクト(<u>P</u>)	ビルド(<u>B</u>) デバッグ(<u>D</u>) ツール(<u>T</u>) ウインドウ(<u>W</u>)	ヘルプ(円) 🧔 🧔
🚳 スタート(S) 📮 🍟 🔏 🖻 🖉	(2) 🛱 🖗 🖣 👘	
	ソリューション一覧(<u>S</u>)	クリックして
🖏 לבטיבלא 🖓 🗘	プロパティ	パスを設定
<u>ک</u> کے <u>ک</u> ک	🔨 CC-RX のプロパティ	
	 入力 オブジェクト・モジュール・ファイル 使用するライブラリ・ファイル システム・ライブラリ・ファイル メバイナリ・データ・ファイル メバイナリ・データ・ファイル シンボル定義 実行開始20下しえを指定する ブレリンカを起動する 出力 	オブジェクト・モジュール・ファイル[0] (使用するライブラリ・ファイル[2]] システム・ライブラリ・ファイル[0] バイナリ・データ・ファイル[0] シンボル定義[0] いいえ 自動制御
	出力ファイル形式	ロード・モジュール・ファイル(-FOrm=Absolute)
Application Config GUI	使用するライブラリ・ファイル ライブラリ・ファイルを指定します。 リンカのオプシュンーIbraryに相当します。 主に次のブレースホルダに対応しています。…	
ia	∖共通オプション Ҳ コンパイル・オプ… Ҳ アセンブル・オプ	<u>、リンク・オプション</u> へキサ出力オプ (ライブラリ・ジェネ/▼

コピーして新しいパスを設定します。

パス編集	×	
パス(1行につき1つのパス)(P): WProjectFolder%/////U: WProjectFolder%/////U:	Electric contraction for the series of the s	%ProjectDir%/GUI/GUILib.lib %ProjectDir%/r_tfat_rx/lib/tfat_rx600_little.lib
参照(B)存在しないパスを許可するプレースホルダ(L):	5(N)	
プレースホルダ ActiveProjectDir ActiveProjectMicomName ActiveProjectName く	値 C:¥Workspace¥RX¥emWin-RX-220603¥emV R5F565NED×FB_DUAL emWin_RX65N > OK キャンセル ヘルプ(H)	

● Step8: RX65N 以外のターゲットボード関連のコードをプロジェクトから削除します

「emWin_RX65N¥r_dmaca_rx¥src¥targets」フォルダ下の以下のターゲットボード関連のソースをプロジェクトビルドから削除します。

- emWin_RX65N¥r_dmaca_rx¥src¥targets¥rx64m
- emWin_RX65N¥r_dmaca_rx¥src¥targets¥rx71m
- emWin_RX65N¥r_dmaca_rx¥src¥targets¥rx230
- emWin_RX65N¥r_dmaca_rx¥src¥targets¥rx231



🕼 emWin_RX65N - CS+ for CC - [プロ	コジ	ェクト・ツリー]
ファイル(F) 編集(E) 表示(V) プロジ	<i>σ</i> τ/	ͻト(Ρ) ビルド(Β) デバッグ(D) ツール(T) ウインドウ(W) ヘルプ(H)
🚳 スタート(S) 🔒 🗐 🐰	b	🗈 🕫 🕶 🔠 🗸 🔍 🔹 100% 👻 🗑 🖓 HardwareDebug 🔹 🔨 🖏 🖏 🗠 🍋 🔘
	2	🗗 🧧 ソリューション一覧(S)
💷 プロジェクト・ツリー		サ× アンプロパティー
🖻 🙎 🕜 🙎 پر		
√ ↓ ∎∎ r_config		
i i r_dmaca_rx		ビルドの対象とする は い
y		ファイルボシント しき語シース しき語シース 個別コンパイル・オブションを設定する いいえ
✓ Ist readme.txt		
r dmaca rx.	.c	
	priva	ate.h
and targets		
i nx231		
tar a star a st		ドルドの対象をする
		Chi.orange 2.2
tr flash rx		追加(D)
	21	
i r_gpio_rx		Windowsエクスプローラのメニュー(W)
in r_irq_rx	_	
	6	プロジェクトから外す(R) Shift+Delete
	D.	그년-(C) Ctrl+C
B- r tfat driver rx	Ē.	貼り付け(P) Ctrl+V
⊕ ∎ r_tfat_rx	aje	名前の変更(N) F2
🗊 🗐 r_usb_basic		
r_usb_hmsc		ノーハン コ (r) 、 9 / (ひ) × * fr ルト・ツール 人 * デビッド・ビルド /
src		

● Step9:不要サンプルをビルド対象外に設定します

「MT_ScratchPad」及び「SlideDemo」プロジェクトをビルド対象外に設定します。







● Step10:プロジェクトを保存→ビルドして設定を確認します



出力 4	д ;	×
<pre>>r_sci_iic_rx#src#targets#rx24t#r_sci_iic_rx24t.c,d >r_dmaca_rx#src#t_dmaca_rx.c,d >r_bsptboard#generic_rx85n#resetprg.c,d >r_sci_iic_rx#src#targets#rx63n#r_sci_iic_rx63n.c,d >r_scl_iic_rx#src#targets#rx65n#r_sci_iic_rx63n.c,d >r_scl_ix=rxd#river#r_usb_petable.c,d >r_scl_ix=rxd#river#r_usb_petable.c,d >r_dmaca_rx#src#targets#rx65n#r_dmaca_rx_target.c,d >r_usb_basic#src#driver#r_usb_fhubsys.c,d >r_usb_basic#src#driver#r_usb_fhubsys.c,d >r_usb_basic#src#driver#r_usb_fhubsys.c,d >r_usb_basic#src#driver#r_usb_fhubsys.c,d >r_usb_basic#src#driver#r_usb_fhubsys.c,d >r_usb_basic#src#driver#r_usb_fhubsys.c,d >r_usb_basic#src#driver#r_usb_fhubsys.c,d >r_usb_basic#src#driver#r_usb_fhubsys.c,d >r_usb_basic#src#driver#r_usb_fnubget.c,d >r_usb_basic#src#driver#r_usb_fnubder_usbip0.c,d >r_usb_basic#src#fur#r_usb_rx_mcu.c,d >r_usb_hasc#src#r_usb_hstorage_driver_api.c,d >src#firm_update.c,d >src#firm_update.c,d >src#firm_upb_hasc_api.c,d</pre>		~
Alter Lob Jinds Labres and Linker Completed」 Renesas Optimizing Linker Completed」 ビルド終了(エラー:0個,警告:5個)(enWin RX65N, HardwareDebug)」 ビルド終了(エラー:0個,警告:5個)(enWin RX65N, HardwareDebug)」 ビルド終了(エラー:0個,警告:5個)(enWin RX65N, HardwareDebug)」 ビルド終了(エラー:0個,警告:5個)(enWin RX65N, HardwareDebug)」 ビルド終了(エラー:0個,警告:5個)(enWin RX65N, HardwareDebug)」 		~ •



45/48

3.4.3. デバッグ設定及びダウンロード・デバッグ操作

● Step11:E2 Lite デバッグ・ツールを設定します

[デバッグ]→ [使用するデバッグ・ツール]コマンドで「RX E2 Lite」を選択します。

Ć) emWin_RX65N - CS+ for CC - [プロジェクト・ツリー]								
ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D) ツール(T) ウインドウ(W) ヘルプ(H)									
🔅 🚳 スタート(S) 退 🔚 🎒 🧮 🛣 🛍 😕			デバッグ・ツール ヘダウンロード(D)		der.	🚮 HardwareDebu	ig 🖣		
i ę	7 🖓 🔊 🧐 🖓 i 🗆 🗭 🗣 🔍 1 5 i i	6	ビルド&デバッグ・ツール ヘダウンロード(B)	F6					
	プロジェクト・ツリー	5	リビルド&デバッグ・ツール ヘダウンロード(W	0					
	2 🕜 🤮 🗷	00	デバッグ・ツールへ接続(C)						
スマート・マニュア		ď	デバッグ・ツールからアップロード(U)						
		J.	デバッグ・ツールから切断(N)	Shift+F6			1024		
			使用するデバッグ・ツール(L)	•		RX E2(2)	E.I.I.		
¥			停止(S)	Shift+F5		RX E2 Lite(E)	e-endiar		
	<u>⊢</u>		実行(G)	F5		RX E1(Serial)(L)			
	■…ŢIJ ヒルト・ツール生成 Jァイル	D	ブレークせずに 実行(E)	F8		RX E1(JTAG)(G)			
		9±	ステップ・イン(1)	F11		RX E20(Serial)(R)			
	MT_ScratchPad	ÇΞ	ステップ・オーバー(O)	F10		RX E20(JTAG)(T)			
		ē_	リターン・アウト(R)	Shift+F11	~	RX シミュレータ(S)			
	in GUI	μ	CPUリセット(T)	Ctrl+F5	771	山設定 / フック処理	調定 /		
	ia∎ drw_2d	H-1	リスタート(A)			Nuch Northers	LUXAL /		

「RX E2 Lite」デバッグ・ツールの「接続用設定」タブの以下の項目設定を変更します。

メイン・クロック	27. 000
起動バンクを変更する	はい
起動バンク	バンク1



R	RX E2 Lite のプロパティ					
	内蔵 RAMサイズ[Kバイト]	640				
	データフラッシュ・メモリ・サイズ[Kバイト]	32				
\sim	<i>ካ</i> ロック					
	メイン・クロック・ソース	EXTAL				
	メイン・クロック周波数[MHz]	27.0000				
	動作周波数[MHz]					
	内蔵フラッシュ・メモリ書き換え時のクロック操作を許可する	ບເນັ				
\sim	Iミュレータとの接続					
	エミュレータシリアル No.					
\sim	ターケット・ボードとの接続					
	エミュレータから電源供給をする(最大200mA)	ບເນັ				
	通信方式	JTAG				
	JTAGクロック[MHz]	6.00				
\sim	7 5 992					
	IDコードの入力モード	IDコードを16進32桁で指定				
	IDコード	HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF				
	ワークRAM開始アドレス	HEX 1000				
	ワークRAMサイズDドイト]	1280				
~	CPUの動作モード					
	モード端子設定	シングルチップモード				
	レジスタ設定	シングルチップモード				
	エンディアン	Little-endianデータ				
	起動バンクを変更する	はい				
	起動バンク	バンクリ				
×	外部フラッシュ					
>	外部フラッシュ定義ファイル	[4]				

Step12:ダウンロードしてデバッグします

[デバッグ]→[ビルド&デバッグ・ツールへダウンロード]コマンドでプログラムをターゲットフラッシュにロ ードしてデバッグステップ・実行操作を確認します。







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 へ通常価格の割引価格でアップグレード可能です。

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



代理店販売(デバッガ、RTOS、GUI、ミドルウエア) https://www.embitek.co.jp

> 日本代理店 SEGGER Microcontroller GmbH



〒130-0021 東京都墨田区緑 4-8-8 中井ビル 4F Phone: 03-6240-2655 Fax: 03-6240-2656

