

# RXファミリ デュアルモードの使用ガイド

# 要旨

本アプリケーションノートは、主にRX671グループの2Mバイト製品を例にデュアルモードの概要、事前 準備、デバッグ方法およびRenesas Flash Programmer (以下、RFPとする)での書き込み方法について説明 します。

RX671グループの2Mバイト製品は、出荷状態ではバンク0のアドレスがFFF0 0000h~FFFF FFFFh、バンク1のアドレスがFFE0 0000h~FFEF FFFFhとなります。使用するマイコンに応じて、本文中のアドレスを読み替えてください。

対象デバイス

RX26Tグループ RX651グループ RX65Nグループ RX66Nグループ RX671グループ RX72Mグループ RX72Nグループ

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



# 目次

1.	概要	3
1.1	デュアルモードとは	3
1.2	ファームウェアアップデート	4
2.	事前準備	5
2.1	バンクの位置が分からない場合	5
2.1.1	│ オプション設定メモリを読み出す方法	5
2.1.2	2 マイコンを出荷状態に戻す方法	9
3.	デバッグの方法	12
3.1	ユーザプログラム Ver1.00 を開発する場合	
3.1.1	プロジェクトの作成手順	
3.1.2	2 BSP の設定手順	15
3.1.3	3 デバッグ構成の設定手順	
3.2	ユーザプログラム Ver.1.01 以降を開発する場合	
3.2.1	プロジェクトの作成手順	
3.2.2	2 BSP の設定手順	23
3.2.3	3 デバッグ構成の設定手順	24
3.3	起動バンク切り替え機能のデバッグ方法	
3.4	ユーザプログラムの mot ファイルを出力する方法	
3.4.1	mot ファイルの出力手順	
3.4.1	I.1 Renesas Electronics C/C++ Compiler Package for RX Family を使用する場合	
3.4.1	I.2 GCC for Renesas RX を使用する場合	
3.4.2	2 オフセットした mot ファイルの出力手順	
4.	RFP を使ったデュアルモードのプログラム書き込み方法	
4.1	出荷状態のマイコンのバンク0に書く方法	
4.2	バンク1に書く方法	
5.	注意事項	45
5.1	オプション設定メモリの設定	
6.	参考ドキュメント	46
改訂	- 「記録	47



### 1. 概要

## 1.1 デュアルモードとは

バンクモード切り替え機能を使用し、リニアモードとデュアルモードの2つのモードを選択可能です。 デュアルモードに切り替えた場合、コードフラッシュメモリのユーザ領域を2つのバンク領域として扱いま す。

起動バンク選択機能によって、プログラムを起動するバンク領域を選択することで、リセットなどによる 書き換え動作の中断に対して安全なプログラム更新方法を提供します。デュアルモードで起動バンク切り替 えした際の書き込みアドレスについて、「図 1.1 起動バンク切り替え時の配置アドレスの例」に示しま す。

オプション設定メモリのBANKSEL.BANKSWP[2:0]ビットの値を設定してからリセットすることで、バン ク1とバンク0のアドレスが切り替わり、FFF00000h~FFFFFFFhに配置されたプログラムが実行されま す。起動バンク選択によりアドレス切り替えを行った状態では、FACIコマンドを発行する対象アドレスも 入れ替わります。例えばBANKSEL.BANKSWP[2:0]ビットが"000b"の場合、バンク0へのFACIコマンドの処 理対象領域は、FFE00000h~FFEFFFFhとなります。起動バンク選択機能はリニアモード選択時には無 効です。



図 1.1 起動バンク切り替え時の配置アドレスの例

1.2 ファームウェアアップデート

デュアルバンク機能を用いたファームウェアアップデートの動作を以下に説明します。

ファームウェアアップデートにデュアルバンク機能を用いることにより、リセットなどによる書き換え動 作の中断に対して、安全に実行することが可能です。ただし、使用できるコードフラッシュメモリの容量は 半分となってしまうため、注意が必要です。

図 1.2にファームウェアアップデートの動作概要を示します。



図 1.2 ファームウェアアップデートの動作概要

- (1) 状態1は出荷状態のマイコンになります。バンク0にユーザプログラム1をフラッシュライタ等で書き
   込むことにより、状態2になります。
- (2) ユーザプログラム2を開発後、状態2のバンク0にあるフラッシュ書き換えプログラムで、ユーザプロ グラム2をバンク1に書き込むと状態3になります。(注1)
- (3) フラッシュ書き換えプログラムで BANKSWP[2:0]ビットを切り替えた後にマイコンをリセットすること により、状態4になります。(BANKSWP[2:0]ビットの切り替えは RAM 上で実行してください)
- (4) 状態 4 ではバンク 1 にあるユーザプログラム 2 が実行されます。その際にバンク 0 にあるユーザプログ ラム 1 を消去することにより、状態 5 になります。
- (5) さらに新しいユーザプログラムを開発後、同様の処理を繰り返します。状態8と状態2は同様の状態と 読み替えてください。
- 注1. 意図しない電源断やリセットなどにより書き換えが失敗した場合でも、BANKSWP[2:0]ビットを切り替えていないため、次回起動時はバンク0のユーザプログラム1が動作します。バンク1を全消去し、再度(2)を実行してください。

## 2. 事前準備

#### 2.1 バンクの位置が分からない場合

バンク0とバンク1のアドレス位置が分からない場合、RFPを使用してオプション設定メモリの値を読み出 して判断する方法と、オプション設定メモリとすべてのフラッシュメモリを消去することにより、出荷状態 に戻す方法で解決することができます。オプション設定メモリの値を読み出す方法は「2.1.1 オプション設 定メモリを読み出す方法」に、オプション設定メモリとフラッシュメモリを消去する方法は「2.1.2 マイコ ンを出荷状態に戻す方法」に示します。

なお、RFP はマイコンのデュアルモードとリニアモードを異なるマイコンとして認識するため、マイコンのバンクモードと異なるプロジェクトで接続した場合、エラーが発生します。新規プロジェクト作成時、 接続するマイコンのバンクモードに応じて、プロジェクト名に「\_dual」や「\_linear」を付加することを推奨します。初めてマイコンと接続したとき、マイコンのバンクモードの情報がプロジェクトに保存されます。

- 2.1.1 オプション設定メモリを読み出す方法
- (1) RFP を起動してください。
- (2)「ファイル(F)」タブをクリックしてください。
- (3)「新しいプロジェクトを作成(N)」をクリックし、デュアルモードのプロジェクトを作成してください。

Manual Renesas Flash Programmer V3.11.02 ファイル(F) へルブ(H)	「ファイル(F)」タブをクリック	- 🗆 X
新しいプロジェクトを作成(N) プロジェクトを開く(O) プロジェクトを保存(S) イメージファイルを保存(I) ファイルチェックサム(C) ファイルパスワード設定(P)	「新しいプロジェクトを	作成(N)」をクリック 参照 (B)
フラッシュ操作 スタ	— þ(S)	
Renesas Flash Programmer V3.11.02 [3 Ap	r 2023]	
		ステータスとメッセージのクリア(C)



- (4)「プロジェクト名(N)」には任意のプロジェクト名を入力してください。
- (5) その他設定は、システムに応じて変更してください。
- (6)「接続(O)」をクリックしてください。

🌠 新しいプロジェクトの作	成	_		×
プロジェクト情報	╱任意のプロ	ジェク	フト名を	入力
マイクロコントローラ(M):	RX67x ~			
プロジェクト名(N):	rx671_dual	]		
作成場所(F):	C:¥workspace¥Renesas Flash Programmer¥V3.11	参	·照(B)	
通信 ツール(T): E2 emulat ツール詳細(D)	or Lite 〜 インタフェース(I): 2 wire UART 〜 番号: 自動選択 電源: 供給しない 「接続(O)」をクリック 接続(O	)	キャンセノ	ih(C)

- (7)「ID コード(I)」には設定しているセキュリティ ID を入力してください。
- (8)「OK」をクリックしてください。

💋 認証	セキュリティ ID を入力 🚽 📃 🛛 🗙
認証コード	
IDコード(I):	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
☑ 自動認証	OK」をクリック OK キャンセル(C)

(9)「ターゲットデバイス(D)」タブから「メモリを読み出す(M)」をクリックしてください。

📕 Renesas	s Flash Programmer V3.11.02	ーゲットデバイス(D)」をクリック				
ファイル(F)	ターゲットデバイス(D) ベルフ(H)					
操作 操作	メモリを読み出す(M)					
	ブロック情報を読み出す(B)	「メモリを読み出す(M)」をクリック				
- フロシェク 理士の	ブランクチェック(C)					
現在の マイクロ	אפצעטרע (אפצעטרע אין אפצעטרע אין אפצעטרע אין אפצעטרעבן אין איז אפצעטרעבן אין איז איז אפצעטערען איז איז איז איז די איז איז איז איז איז איז איז איז איז אי	エンディアン(E): リトル 🗸				
プログラム	บวราน					
		李炽((B)				
フラッシュ	操作					
消去ン	消去 >> 書き込み >> ベリファイ					
スタート(S)						



(10)保存する任意のフォルダに移動してください。

(11)「ファイル名(N)」には任意のファイル名を入力してください。

(12)「保存(S)」をクリックしてください。

🌠 名前を付けて保存				×
$\leftarrow \rightarrow$ $\checkmark$ $\uparrow$ $\blacktriangleright$ > PC > Windows (C:)	workspace 任	意のフォルダに移	<mark>動</mark> , iaceの検索	
整理 ▼ 新しいフォルダー			≣≡≣ ▾	?
名前 ^	更新日時	種類	サイズ	
e2_studio Renesas Flash Programmer	2023/05/29 14:55 2023/05/19 16:14 千音のファイルタを入っ	ファイル フォルダー ファイル フォルダー		
ファイル名(N): ofsm		J		~
ファイルの種類(T): Sレコードファイル (*.mot)				$\sim$
「保ィ へ フォルダーの非表示	存(S)」をクリック —	保存(	S) キャンセル	]

(13)「エリアの選択(A)」から「Config Area」を選択してください。

(14)「読み出し(R)」をクリックしてください。

📕 メモリ情報の読み出し			_		×
エリアの選択(A)	Config Area	を選択	)設定		
All Areas Code Flash 1 Data Flash 1		開始 FE7F	アドレス(約 5000	5): [HEX]	
Config Area		終了	アドレス(ド	E): [HEX]	
		FE7F	5D7F		
開始アドレス = 0×FE7F5D0 終了アドレス = 0×FE7F5D7	0 F	データ	2長(L): [J	HEX]	
サイズ = 0.13 K		80			
「読みb	出し(R)」をクリ	ック			
🗹 ブランク領域をスキップして	[リード(P)				
		読み出	J(R)	Can	cel



(15)「操作が成功しました。」と表示されると完了です。(12)で指定した mot ファイルからオプション設定 メモリを読み出すことができます。

Renesas Flash Programmer V3.11.02		_		$\times$
ファイル(F) ターゲットデバイス(D) ヘルプ(H)				
操作 操作設定 ブロック設定 接続設定 ユニークコード				
プロジェクト情報 現在のプロジェクト: rx671_dualrpj マイクロコントローラ: RX Group エンデ・ プログラムファイル	ィアン(E):	リトル 参	~ 照(B)	]
フラッシュ操作 <sub>「34+</sub>				
スタート(S)		正常約	終了	
通信速度:1,000,000 bps Signature: Device: RX Group 読込みを実行します。 [Config Area] 0xFE7F5D00 - 0xFE7F5D7F サイズ:128				^
<u>ッールから切断します。</u> 操作が成功しました。 「操作が成功しました。」と表示される				>
ſ	ステータ	マンとメッセ	ージのクリン	7(C)



2.1.2 マイコンを出荷状態に戻す方法

アクセスウィンドウを設定している場合は、事前に設定を解除する必要があります。詳細は使用するマイ コンのユーザーズマニュアルを参照してください。

- (1) RFP を起動してください。
- (2)「ファイル(F)」タブをクリックしてください。
- (3)「新しいプロジェクトを作成(N)」をクリックし、デュアルモードのプロジェクトを作成してください。

📕 Renesas Flash Programmer V3.11.02	「ファイル(F)」タブをクリック ー 🗆 🗙
ファイル(F) <u>ヘルノ(H)</u>	
新しいプロジェクトを作成(N)	
<b>プロジェクトを開く(O)</b> プロジェクトを保存(S)	「新しいプロジェクトを作成(N)」をクリック
イメージファイルを保存(I) ファイルチェックサム(C)	
ファイルパスワード設定(P)	
終了(X)	参照(B)
フラッシュ操作 スタ	-

- (4)「プロジェクト名(N)」には任意のプロジェクト名を入力してください。
- (5) その他設定は、システムに応じて変更してください。
- (6)「接続(O)」をクリックしてください。

🌠 新しいプロジェクトの作り	戎	_		×
プロジェクト情報	任意の	のプロジェ?	フト名を	入力
マイクロコントローラ(M):	RX67x ~	/		
プロジェクト名(N):	rx671_dual			
作成場所(F):	C:¥workspace¥Renesas Flash Programmer¥\	V3.11 参	照(B)	]
通信 ツール(T): E2 emulato ツール詳細(D)	or Lite 〜 インタフェース(I): 2 wire UART 〜 番号: 自動選択 電源: 供給しない 接	「 <mark>接続(O)」</mark> 蒜(O)	<b>をクリ</b> ッ キャンセノ	ν <b>ク</b> μ(c)



(7)「ID コード(I)」には設定しているセキュリティ ID を入力してください。

(8)「OK」をクリックしてください。

🛃 認証	セキュリティ ID を入力 🛑 🗌 🛛 🗙
認証コード	
IDコード(I):	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
「 <mark>⊂</mark> 」 「」 「」 「」	OK」をクリック OK キャンセル(C)

(9)「操作設定」タブをクリックしてください。

(10)「コマンド」は「消去(E)」のみチェックを入れてください。

(11)「消去オプション(O)」で「チップ消去」を選択してください。

Renesas Flash Programmer V3.11.02	-
ファイル(F) ターゲットデバイス(D) トリーイル (F) ターゲットデバイス(D) 「操作設定」タブを 操作 操作設定 フロック設定 接続設定 ユニークコード	クリック 「チップ消去」を選択
<ul> <li>コマンド</li> <li></li></ul>	消去オブション(0) チップ消去 書き込みとベリファイオブション 日書き込み前に消去(B) デバイスからリードしてベリファイ ~
「消去(E)」のみチェックを入れる	チェックサム計算方式(M) 32bit 加算方式 ~
0xFF補完 Code Flash / User Boot Data Flash	エラー設定 「ビ デバイス範囲外エラーを有効にする(R)

(12)「操作」タブを開き、「スタート(S)」をクリックしてください。

(13)「操作が成功しました。」と表示されると完了です。

🕻 Renesas Flash Programmer V3.11.02		_		$\times$
ファイル(F) ターゲットデバイス(D) ヘルプ(H)				
操作 操作設定 ブロック設定 接続設定 ユニークコード 「操作」タブをクリック				
現在のプロジェクト: rx671_dualrpj マイクロコントローラ: RX Group エン	ディアン(E)	። ሀኑル	~	
プログラムファイル			参照(B)	
コンシュナポルド コスタート(S)」をクリック 消去				-
スタート(S)」をクリック 消去 スタート(S)		正常	終了	
フラッシュ採作     「スタート(S)」をクリック       消去     スタート(S)       選択されたブロックを消去します。     [Data Flash 1] 0x00100000 - 0x00101FFF サイズ:8 K       [Code Flash 1] 0xFFE00000 - 0xFFFFFFFF サイズ:2.0 M		正常	終了	^
フラッシュ採作     「スタート(S)」をクリック       消去     スタート(S)」       選択されたブロックを消去します。     [Data Flash 1] 0×00100000 - 0×00101FFF サイズ:8 K       [Code Flash 1] 0×FFE000000 - 0×FFFFFFFF サイズ:2.0 M       選択されたブロックを消去します。       [Config Area]		正常	終了	^
フラッシュ採作       「スタート(S)」をクリック         消去       スタート(S)         選択されたブロックを消去します。       [Data Flash 1] 0x00100000 - 0x00101FFF サイズ:8 K         [Code Flash 1] 0xFFE00000 - 0xFFFFFFFF サイズ:2.0 M         選択されたブロックを消去します。         [Config Area]         ツールから切断します。         「操作が成功しました。」と表示され	₫	正常	終了	

3. デバッグの方法

e<sup>2</sup> studio を用いた、デュアルモードについてのデバッグ方法を示します。

ユーザプログラムの開発及びデバッグは、ユーザプログラムを書き込むアドレスに関わらず、実行領域である FFF0 0000h~FFFF FFFFh に配置して実施してください。

ユーザプログラムを新規開発する場合は、「3.1 ユーザプログラム Ver1.00 を開発する場合」を参照して ください。ユーザプログラムをアップデートする場合は、「3.2 ユーザプログラム Ver.1.01 以降を開発する 場合」を参照してください。

3.1 ユーザプログラムVer1.00を開発する場合

ユーザプログラムのプロジェクトを新規作成し、Borad Support Packages (以下、BSP とする)とデバッ グの構成を設定する手順を説明します。

3.1.1 にプロジェクトの作成手順、3.1.2 に BSP の設定手順、3.1.3 にデバッグ構成の設定手順を示します。

3.1.1 プロジェクトの作成手順

(1) e<sup>2</sup> studio を起動してください。

(2)「ファイル(F)」タブをクリックしてください。

(3)「新規(N)」→「Renesas C/C++ Project」→「Renesas RX」の順にクリックしてください。

771	(ル(F)	Alt+シフト+N >		Renesas Views 実 Renesas C/C++ Project	行(R) ウィンドウ(W) / >	Nレプ(H) Renesas Debug
۵,	ファイルを開く、 ファイル・システムやらプロジェクトを開く 最近のファイル	>	یم ۲۵ ۲	Makefile Project with Existing Code C/C++ Project プロジェクト(R)	[	Renesas RL/8 Renesas RX
	<sup>閉じる(C)</sup> 「新規(N)」をクリ すべて閉じ	ック <sup>Ct+1</sup>	es	as C/C++ Project」をクリ	ック	
	保存(S) 名前を付けて保存(A) すべて保存(E) 前回保存した状態に戻す(T)	Ctrl+S Ctrl+シフト+S	°	フォルダー ソース・ファイル ヘッダー・ファイル テンプレートからファイル	「Renesas R)	/ (」をクリック
<b>8</b>	移動(V) 名前を変更(M) 更新(F) 行区切り文字の変換(D)	F2 F5		クラス コード生成 その他(o)	Ctrl+N	
₽	印刷(P)	Ctrl+P				
24 24	インボート(I) エクスポート(O)					
	プロパティ(R)	Alt+Enter				
	ワークスペースの切り替え(W) 再開 終了(X)	>				



- (4) 使用するコンパイラに合わせて「GCC for Renesas RX C/C++ Executable Project」もしくは「Renesas CC-RX C/C++ Executable Project」をクリックしてください(今回は Renesas CC-RX を使用します)。
- (5)「次へ(N)」をクリックしてください。

New C/C++	+ Project			×
GCC fo	or Renesas RX」を使用する場合は			
GCC fo	or Renesas RX C/C++ Executable Project」をク	リック		
All	GCC for Renesas RX C/C++ Executable Project			
C/C++	A C/C++ Executable Project for Renesas RX using the G	<i>CC for Renesa</i> :	s RX Toolc	hain.
	GCC for Renesas RX C/C++ Library Project			
	A C/C++ Library Project for Renesas RX using the GCC f	for Renesas RX	Toolchain	ł
	Renesas CC-RX C/C++ Executable Project			
	A C/C++ Project for Renesas RX using the Renesas CC-I	RX toolchain.		
	Renesas CC-PX C/C++ Library Project		a a.	
	A C/C++ Library Project for Renesas RX using the Renes	sas CC-RX tool	chain.	
[Renes	。 as CC-RX」を使用する場合は as CC-RX C/C++ Executable Project」をクリッ	л		>
· Kenese				
[*#Z @				
?	<(N)」をワリック < 戻る(B) 次へ(N) >	終了(F)	キャンセ	

- (6)「プロジェクト名(P)」に任意のプロジェクト名を入力してください。
- (7) その他の設定は、システムに応じて変更してください。
- (8)「次へ(N)」をクリックしてください。

8			×
New Renesas CC-RX New Renesas CC-RX Ex	Executable Project 任意のプロジェクト名をJ recutable Project	እታ 📒	Ź
プロジェクト名(P): firmw	are1		
☑ デフォルト・ロケーション	の使用(D)		
ロケーション(L):	C:¥workspace¥e2_studio¥firmware1	参照(R)	)
	Create Directory for Project		
ファイル・システムを選択(Y	): デフォルト \vee		
ワーキング・セット			
□ ワーキング・セットにプロ	コジェクトを追加(T)	新規(W).	
ワーキング・セット(O):	~	選択(E)	
「次へ(N)」を?	フリック		
?	< 戻る(B) 次へ(N) > 終了(F)	キャンさ	zIL



- (9)「Hardware Debug 構成を生成」にチェックを入れてください。
- (10) その他の設定は、システムに応じて変更してください。
- (11)「次へ(N)」をクリックしてください。

6			- 🗆 X
New Renesas CC-RX Select toolchain, device	C Executable Project		
Toolchain Settings 言語: ツールチェーン: ツールチェーン・パージョン: RTOS: RTOS Version: 「H	● C ○ C++ Renesas CC-RX v3.05.00 None ardware Debug 構品	〜 〜 <u>ツールチェーンの管理</u> ↓ 戈を生成」にチェッ	クを入れる
Device Settings			Configurations
Target Board: Cus	stom	Download additional boards	✓ Hardware Debug 構成を生成 E2 Lite (RX) ✓
ターゲット・デバイス: R5I	F5671EDxFB_DUAL		Debug 構成を生成 RX Simulator ~
プロジェクト・タイプ: デフ	オルト	~	<ul> <li>Release 構成を生成</li> </ul>
「次~	ヽ(N)」をクリック 、		
?	[	< 戻る(B) 次へ(N) >	終了(F) <b>キャンセル</b>

- (12)「Use Smart Configurator」にチェックを入れてください。
- (13)「終了(F)」をクリックすると、プロジェクトの作成は完了です。





- 3.1.2 BSP の設定手順
- (1) スマートコンフィグレータを開いてください。
- (2)「コンポーネント」タブをクリックしてください。
- (3)「r\_bsp」をクリックしてください。
- (4)「Select the startup bank」は「Bank1:FFE00000h~FFEFFFFh; Bank0: FFF00000h~FFFFFFFh.」を 選択してください。
- (5) その他の設定は、システムに応じて変更してください。
- (6)「コードの生成」をクリックすると、BSPの内容がソースコードに反映されます。
- (7) ビルドを実施してください。以上で BSP の設定は完了です。





3.1.3 デバッグ構成の設定手順

(1)「プロジェクト・エクスプローラー」上で設定するプロジェクトをクリックしてください。

e2_studio - e <sup>2</sup> studio	
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 Views 実行(
参 またまでは、 ないからの ない	✓ Irmware2 HardwareDe
🔁 プロジェクト・エクスプローラー 🗙	プロジェクトをクリック
😂 댥 firmware1 [HardwareDebug]	
> 💕 firmware2	

(2)「実行(R)」タブをクリックし、「デバッグの構成(B)」を選択してください。

a views	実行	<del>〕(R)</del> ウィンドウ(W)	ヘルプ(H)		
1 Hardv	略	実行(R) デバッグ(D)	「実行(R)」タブ	をクリッ	ク 11
0	0	実行履歴(T) 実行(S)		>	
	**	実行構成(F デバッグ変換	、ッグの構成(B)	」をクリ	ック
	200	デバッグの構成(B)			
	0	外部ツール(E)		>	

- (3)「Debugger」タブ内の「Connection Settings」タブをクリックしてください。
- (4)「起動バンクを変更する」を「はい」に変更してください。
- (5)「起動バンク」を「バンク0」に変更してください。

国 デバッグ構成 構成の作成、管理、および実行		_ 「Debugger」タブをクリック	
			)Ôr
C 🛙 🕼 🗮 🗮 🖻 🍸 🔹	名前(N): firmware1 HardwareDebug	Connection Settings	タブをクリ
フィルタ入力	📄 メイン 🐯 Debugger 🕨 Startup 🔲 共	₫(C) V-ス	
<ul> <li>C/C++ アプリケ−ション</li> <li>C/C++ リモート・アプリケ−ション</li> <li>EASE Script</li> </ul>	Debug hardware: E2 Lite (RX) ~	larget Device: R5F5671E_DUAL	
GDB Simulator Debugging (RH850)	GDB Settings Connection Settings デパック	ブ・ツール設定	
GDB ハートウェア・デバッキンク	エミュレーター	(Auto)	^
Java アブリケーション	接続タイプ	JTag	~
Java PJUMF	JTag クロック周波数[MHz]	6.00	~
Renesas GDB Hardware Debugging	Fine ボーレート[Mbps]	1.50	~
c firmware1 HardwareDebug	ホット・プラグ	いいえ	~
firmware2 HardwareDebug	✓ 電源		
Renesas Simulator Debuggi	*	いいえ	~
望 リモート Java アブリケーション 「 正  IJ	シッを変更する」を「はい」	-変更   3.3	~
🖏 起動グループ	✓ 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	v
	起動バンクを変更する	はい	~
	起動バンク	バンクロ	~
	◇ 通信モード		
		デバッグ・モード	~
	「起動ハンク」を「ハンク 0」	に変史 いいえ	~
	◇ フラッシュ		
	10.7-5		CEC



(6)「デバッグ・ツール設定」タブをクリックしてください。

(7)「内蔵プログラム ROM を書き換えるプログラムをデバッグする」を「はい」に変更してください。
 (8)「適用(Y)」→「閉じる」の順にクリックしてください。以上でデバッグ構成の設定は完了です。

			19
	名前(N): firmware1 HardwareDebug	デバッグ・ツール設定」タ	ブをクリック
ルタ入力	📄 メイン 🍄 Debugger 🕨 Startup 🦆 ソース [	□ 共通(C)	
<ul> <li>C/C++ アプリケーション</li> <li>C/C++ リモート・アプリケーション</li> <li>■ EASE Script</li> </ul>	Debug hardware: E2 Lite (RX) ~ Targ	et Device: R5F56718_DUAL	
GDB Simulator Debugging (RH850)	GDB Settings Connection Settings デバッグ・ツ		
GDB ハードウェア・デバッギング	~ 10		^
🗾 Java アプリケーション	デフォルト IO ファイル名を使用	はい	~
🖭 Java アプレット	IO ファイル名	\${support_area_loc}	
Renesas GDB Hardware Debugging	~ 一般		
Firmware1 HardwareDebug	ダウンロード後にリセットする	はい	~
💽 firmware2 HardwareDebug	◇ メモリー		
💽 Renesas Simulator Debugging (RX, RL	エンディアン	リトル・エンディアン	~
🛃 リモート Java アプリケーション	メモリー書き込み時のペリファイ	いいえ	~
	た書も協ってプログニノ	[0]	
「内蔵ノロクラム KOIVI	を書き換えるノログラム	[0]	-
をデバッグする」を「は	い」に変更	0x1000	
	ワーク RAM サイズ (Bytes)	0x500	
	~ システム		
	内蔵プログラムROMを書き換えるプログラムをラ	デバッグする はい	~
	内蔵データ・フラッシュを書き換えるプログラムを	デバッグする いいえ	~
	✓ Start / Stop 機能設定		
	ユーザー・プログラム実行前に関数を実行する	いいえ	~
	start 関数の開始アドレス 「適用(Y)」	をクリック	
>		**回見管  大津龍に	ラオハハ 海田(V)

3.2 ユーザプログラム Ver.1.01 以降を開発する場合

ユーザプログラム Ver1.00 をアップデートするプロジェクトを作成し、BSP とデバッグの構成を設定する 手順を説明します。

3.2.1 にプロジェクトの作成手順を、3.2.2 に BSP の設定手順を、3.2.3 にデバッグ構成の設定手順を示します。

3.2.1 プロジェクトの作成手順

ユーザプログラム Ver1.00 を元にプロジェクトを作成する手順を以下に示します。

(1) e<sup>2</sup> studio を起動してください。

(2)「ファイル」タブから「インポート」をクリックしてください。

	e2_studio - e² stu	udio					
77	イル(F) 編集(E)	ナビゲート(N)	検索(A)	プロジェクト(P)	Ren	esas Views	実行(R)
	新規(N) ファイルを開く(.)		「ファイ	ル(F)」タブ	をク	リック gura	ations
	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				>		
	閉じる(C)			Ctrl+V	N		
	すべて閉じる(L)			Ctrl+シフト+V	N		
	保存(S)			Ctrl+	S		
	名前を付けて保ィ	字(A)					
6	すべて保存(E)	総1-百十(五)		Ctrl+シフト+	S		
	10日休仔した状	膨に戻9(1)					
	移動(V) 名前を恋更(M)			F	2		
รา	互前で変更(₩)。 更新(F)			F	5		
	行区切り文字の	変換(D)			>		
۵	印刷(P)			Ctrl+	Р		
$\geq$	インボート(I)						
	エクスポート(O)		$\overline{}$				
	プロパティ(R)			<b>~</b> [インポー	- ト(	をクリ	ック
	ワークスペースの切	)り替え(W)			>	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	••
	再開						
	終了(X)						

- (3)「一般」から「Rename & Import Existing C/C++ Project into Workspace」をクリックしてください。
- (4)「次へ(N)」をクリックしてください。

<ul> <li>ゴインポート</li> <li>選択</li> </ul>		~_~~		×
Rename and Import and Exist	ting C/C++ Project into the worksp	ace		
インボート・ウィザードの選択(5):	[Rename & Import	Existing	C/C+4	
	Project into Works		しつ ()して () () ()	<b>b</b> 1
71/02/13	Floject into works	harel a	999	
~ № -般		_		^
🕼 Rename & Import	Existing C/C++ Project into Worksp	pace		
Reneas Webサイト上	ニのサンプル・プロジェクト			
Renesas CA78K0R	(CS+) プロジェクト			
Renesas CC-RX/CC	-RL (CS+) プロジェクト			
Renesas CC-RX pro	ject conversion to Renesas GCC RX	(		
🚔 Renesas GitHub Fre	eeRTOS (with IoT libraries) プロジェク	1		
🔎 アーカイブ・ファイル				
😂 ファイル・システム				
😂 フォルダーまたはアーカ	イブ由来のプロジェクト			
1200 既存プロジェクトをワー	クスペースへ			
1. 設定				
> 🧁 C/C++				
> 🧁 Oomph				
5 👝 T				~
「次へ(N)」をクリック				

- (5)「プロジェクト名(P)」に任意のプロジェクト名を入力してください。
- (6)「ルート・ディレクトリーの選択(T)」の「参照(R)」をクリックしてください。

📴 インポート				×
名称変更とプロシェ 既存の Eclipse ブロシ	<b>クトのインボート</b> ジェクトを検索するディレクレクーに	王意のプロジェクト	名を入力	7
プロジェクト名(P): fir	mware2			
☑ デフォルト・ロケーシ	/ヨンの使用(D)			
ロケーション(L):	C:¥workspace¥e2_stue	dio¥firmware2	参照(R)	
	Create Directory fo	r Project		
ファイル・システムを選択	沢(Y): デフォルト ~			
Import from:	「参照(R)」	をクリック		
◉ ルート・ディレクトリ	ーの選択(T):	~	参照(R)	
○ アーカイブ・ファイル	の選択(A):	~	参照(R)	
プロジェクト(P):				
+12,22				
Keep build cont	iguration output folders			



(7) e<sup>2</sup> studio のワークスペースフォルダまで移動してください。

- (8) 元にするユーザプログラムのプロジェクトのフォルダをクリックしてください。
- (9)「フォルダーの選択」をクリックしてください。

フォルダーの選択				×
← → × ↑ 📜 « workspace > e2	2_studio	クスペースフォノ	レダまで移動	
整理▼ 新しいフォルダー			== -	?
名前	更新日時	種類	サイズ	
.metadata	2023/06/13 10:21	ファイル フォルダー		
📜 firmware1	2023/06/13 10:23	ファイル フォルダー		
「フォルダーの選掛	プロジェクト	<b>- のフォルダを</b> ク	リック	
フォルダー: firmware1	$\sim$			

(10)「プロジェクト(P)」から(8)で選択したプロジェクトをクリックしてください。

(11)「終了(F)」をクリックすると、プロジェクトが作成されます。

🖾 インポート		—	×
名称変更とプロシェクト 既存の Eclipse プロジェク	<b>のインボート</b> トを検索するディレクトリーを選択します。		
プロジェクト名(P): firmw	are2		
☑ デフォルト・ロケーション	の使用(D)		
ロケーション(L):	C:¥workspace¥e2_studio¥firmware2	参照(R)	
	Create Directory for Project		
ファイル・システムを選択(Y	): デフォルト 🗠		
Import from:			
◉ ルート・ディレクトリーの	選択(T): C:¥workspace¥e2_studio¥firmware1	✓ 参照(R).	
○ アーカイブ・ファイルの運	银(A):		-
プロジェクト(P):	71919	トをクリッ	
firmware1 (C:¥works	space¥e2_studio¥firmware1)		
オプション			
Keep build configu	ration output folders		
「終了	(F)」をクリック、		
?	< 戻る(B) 次へ(N) > 終了(F)	キャンセ	.JL



作成されたプロジェクトに含まれる、スマートコンフィグレータのファイル名は、自動的に変更されず、 元のプロジェクトの状態になっています。ファイル名を変更する手順を以下に示します。

また、ファイル名が元のプロジェクトの状態でも、動作に影響はありません。以下の手順は任意で実施してください。

(12) ファイル名を変更したいファイルをクリックしてください。

e2_studio - e <sup>2</sup> studio				
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	Viev
🐔 🎋 🔳		~ No	Launch Co	nfig
陷 プロジェクト・エクスプローラー 🗙	🖹 問題	🖻 😫 🍸	000	
> 📂 firmware1				
✓	oug]			
> 🔊 Includes				
> 🐸 src	_			
> 🗁 trash		アイルをクリ	ノック	
firmware1.scfg				
> ⑦ Developer Assistance				

(13)「ファイル(F)」タブから「名前を変更(M)」をクリックしてください。

	e2_studio - e² studio		
771	<mark>イル(F) 編集(E)</mark> ナビゲート(N)	検索(A) プロジェクト(P) Ren	nesas Views 実行(R)
6	新規(N) ファイルを開く(.) ファイル・システムからプロジェクト 最近のファイル	「ファイル(F)」タブをク <sup>-</sup> を開く >	リック <sub>jurations</sub>
	閉じる(C) すべて閉じる(L)	Ctrl+W Ctrl+シフト+W	
	保存(S) 名前を付けて保存(A) すべて保存(E) 前回保存した状態に戻す(T)	Ctrl+S 「名前を変更(M)」	をクリック
	移動(V)	5	
\$	石前を変更(W) 更新(F) 行区切り文字の変換(D)	F5 >	
۵	印刷(P)	Ctrl+P	
24 24	インボート(I) エクスポート(O)		
	プロパティ(R)	Alt+Enter	
	ワークスペースの切り替え(W) 再開 終了(X)	>	
_		,	



(14)「新しい名前(M)」に任意のファイル名を入力してください。

(15)「OK」をクリックしてください。以上でファイル名の変更は完了です。

📴 リソース名の変更	_		×
新しい名前(M): firmware2.scfg			
✓ 参照の更新(R) Open preferences	~任意のファ	イル名を	入力
「OK」をクリック			
プレビュー >(W)	OK	キャンセ	zIV



#### 3.2.2 BSP の設定手順

mot ファイルに BANKSEL.BANKSWP[2:0]ビットの設定を出力しないようにするために vecttbl.c のオプ ション設定メモリのコードを無効にする必要があります。

 (1)「プロジェクト・エクスプローラ」上で「firmware2/src/smc\_gen/r\_bsp/mcu/rx671」のパスにある 「vecttbl.c」をダブルクリックしてください。(パスの「firmware2」はプロジェクト名、「rx671」は使用するマイコンになります)



(2)「vecttbl.c」のコードを変更してください。使用するコンパイラによってコードの変更箇所が異なります。

・Renesas CC-RX の場合

109 行目の「const uint32\_t \_\_BANKSELreg = BSP\_PRV\_START\_BANK\_VALUE;」をコメントアウトしてください。

vecttbl.c ×	
99	<pre>#pragma addressOSIS2reg = 0xFE7F5D54</pre>
100	<pre>#pragma addressOSIS3reg = 0xFE7F5D58</pre>
101	<pre>#pragma addressOSIS4reg = 0xFE7F5D5C</pre>
102	#pragma addressFAWreg = 0xFE7F5D64
103	#pragma addressROMCODEreg = 0xFE7F5D70
104	
105	<pre>const uint32_tMDEreg = (BSP_PRV_MDE_VALUE &amp; BSP_PRV_BANK_MODE_VALUE);</pre>
106	<pre>const uint32_tOFS0reg = BSP_CFG_OFS0 PEG_VALUE;</pre>
107	<pre>const uint32_tOFS1reg = BSP_CFG_OF51_REG_VALUE;</pre>
108	const wint22_tTMINFrog - Oxffffffff;
109	<pre>//const uint32_tBANKSELreg = BSP_PRV_START_BANK_VALUE;</pre>
110	<pre>const uint32_tSPLUreg = BSP_PRV_SPLU_VALUE;</pre>
111	<pre>const uint32_tTMEFreg = BSP_CFG_TRUSTED_MODE_FUNCTION;</pre>
112	<pre>const uint32_tOSIS1reg = BSP_CFG_ID_CODE_LONG_1;</pre>
113	<pre>const uint32_tOSIS2reg = BSP_CFG_ID_CODE_LONG_2;</pre>
114	<pre>const uint32_tOSIS3reg = BSP_CFG_ID_CODE_LONG_3;</pre>
115	<pre>const uint32_tOSIS4reg = BSP_CFG_ID_CODE_LONG_4;</pre>
116	<pre>const uint32_tFAWreg = BSP_CFG_FAW_REG_VALUE;</pre>
117	<pre>const uint32_tROMCODEreg = BSP_CFG_ROMCODE_REG_VALUE;</pre>
118	
119	<pre>#elif defined(GNUC)</pre>

・GCC for Renesas RX の場合

127 行目の「const uint32\_t \_\_BANKSELreg \_\_attribute\_\_ ((section(".ofs3"))) = BSP\_PRV\_START\_BANK\_VALUE;」をコメントアウトしてください。

vecttbl.c ×	
118	
119	<pre>     #elif defined(GNUC) </pre>
120	
121	<pre>const st_ofsm_sec_ofs1_tofsm_sec_ofs1attribute ((section(".ofs1"))) = {</pre>
122	(BSP_PRV_MDE_VALUE & BSP_PRV_BANK_MODE_VALUE), /*MDEreg */
123	BSP_CFG_OFS0_REG_VALUE, /*OFS0reg */
124	BSP_CFG_OFS1_REG_VALUE /*OFS1reg */
125	3:
126	const uint22 + _TMTNErog _attribute_ ((section(" (c2"))) - @vffffffff;
127	<pre>//const uint32_tBANKSELregattribute ((section(".ofs3"))) = BSP_PRV_START_BANK_VALUE;</pre>
128	<pre>const uint32_tSPCCregattribute ((section(".ofs4"))) = BSP_PRV_SPCC_VALUE;</pre>
129	<pre>const uint32_tIMEFregattribute_ ((section(".ofs5"))) = BSP_CFG_IRUSIED_MODE_FUNCTION;</pre>
130	const st_ofsm_sec_ofsb_tofsm_sec_ofsbattribute_ ((section(".ofsb"))) = {
131	BSP_CFG_ID_CODE_LONG_I, /*OSISITEg */
132	BSP_CFG_ID_CODE_LONG_2, /*OSISZREg //
133	BSP_CFG_TD_CODE_LONG_3, /*OSISSING_ */
134	bsp_crd_tb_cobe_come_4 /*oststreg */
135	(section(" of 7"))) - BCD (FG FAW REG VALUE)
137	const uint2_t ((section())) = BSP_CFG_PAM_CBUCKELC,
138	const differz_tmoncosticgetclipate_ ((section( .orso ))) = bsr_crd_moncost_rtd_vALDE,
139	<pre>#elif defined(ICCRX)</pre>

(3) ビルドを実施してください。以上で BSP の設定は完了です。

3.2.3 デバッグ構成の設定手順

「3.1.3 デバッグ構成の設定手順」と同様の手順で設定を行ってください。



3.3 起動バンク切り替え機能のデバッグ方法

バンク0にユーザプログラム1、バンク1にユーザプログラム2をダウンロードし、バンク0とバンク1の切り替えをデバッグする手順の例を以下に示します。

(1)「プロジェクト・エクスプローラー」上でユーザプログラム 1(バンク 0 にダウンロード)のプロジェクトを クリックしてください。

😰 e2_studio - e² studio	
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 Views 実行(
🐔 🏘 🔳 🏘 デバッグ(B)	✓ Ifrmware2 HardwareDe
<u>№</u> ЛОЎІЛ-Э- ×	プロジェクトをクリック
😂 firmware1 [HardwareDebug]	
> 🚰 firmware2	

(2)「実行(R)」タブから「デバッグの構成(B)」をクリックしてください。

s Views	実行	(R) ウィンドウ(W) へ	Jレプ(H)		
1 Hardv 8 🗖	Q。 格	実行(R) デバッグ(D)	钅行(R)」	タブをクリ	ック
0	0	実行履歴(T) 実行(S) 実行構成( <b>)</b> <sup>™</sup>			> >
	<b></b>	デバッグ 展出 デバッグ (G) デバッグの 構成(B)	ッグの構 	成(B)」をク 	・リック →
	<b>Q</b>	外部ツール(E)			>

(3)「Startup」タブを開き、「追加」をクリックしてください。

名前(N): firmware1 HardwareDe メイン 参 Debugger Sta 初期化コマンド ロリセットと遅延(秒): 0	bug rtup te v-z 🔲 # rSta	<sub>も通(C)</sub> artup」タブ	「をクリック	7	^
Halt					
イメージとシンボルをロード					~
ファイル名	ロード・タイプ	オフセット	接続時	追	bд
	1x-2027/1/1/	Chine	t b II w b	-	2017 第七 201
			をクリック	間	去
				Ŀ	-^
				T	5
ランタイム・オブション □ プログラム・カウンター設定先(16	進):				
☑ ブレークポイント設定先:	main				
国国際					~



(4)「ダウンロード・モジュール名の指定」にユーザプログラム 2(バンク 1 にダウンロード)を選択し、「OK」をクリックしてください。

🗐 ダウンロード・モジュールの追加	メードプログラムッを選択
ダウンロード・モジュール名の指定:	
{workspace_loc:¥firmware2¥HardwareD	)ebug¥firmware2.x}
変数 プロジェクトの検索	ワークスペース ファイル・システム
「OK」をクリック	OK キャンセル

- (5) 追加したユーザプログラム2はバンク0(FFF00000h~FFFFFFF)で開発しており、それをバンク 1(FFE00000h~FFEFFFFF)にダウンロードするために「オフセット」の変更が必要です。ここで指 定する値はアドレス値ではなくオフセット値です。RX671グループの2Mバイト製品の場合は、アドレ ス値の差分として-100000Hオフセットします。マイナス値を入力できないため2の補数である FFF00000Hを入力します。
- (6) マイコンに接続し、「デバッグ(D)」をクリックしてください。

📳 デバッグ構成					_		×
構成の作成、管理、および実行						Ř	ñ
<ul> <li> アイルタスカ </li> <li> アイルタスカ </li> <li> C/C++ アブリケ-ション </li> <li> C/C++ リモート・アブリケ-ション </li> <li> EASE Script </li> <li> GDB バードウェア・デバッギング </li> <li> Java アブリケーション </li> <li> Java アブリケーション </li> <li> Java アブリケーション </li> <li> Renesas GDB Hardware Debugging </li> <li> firmware1 HardwareDebug </li> <li> firmware2 HardwareDebug </li> <li> Renesas Simulator Debugging (RX, RL78) </li> <li> Uモート Java アブリケーション </li> <li> 起動グルーブ </li> </ul>	<ul> <li>名前(N): firmware1 HardwareDe</li> <li>メイン 梦 Debugger ▶ Sta</li> <li>初期化コマンド</li> <li>リセットと遅延(秒): 0</li> <li>Halt</li> <li>イメージとシンボルをロード</li> <li>ファイル名</li> <li>ブログラム・パイナリー [firm</li> <li>ダ firmware2.x [C:¥worksp</li> <li>オフセット</li> <li>デンタイム・オブション</li> <li>ブログラム・カウンター設定先(16</li> <li>マブレークボイント設定先:</li> </ul>	bug rtup □ 共通(C) □-ド・タイブ イメージとシンボル イメージとシンボル 直を変更 進): main	> ソース オフセット 0 FFF00000	接続時 Yes Yes	<b>追加</b> 編編 除 上 下	~ ~ ~ ~	
< > 15 項目のうち 13 項目がフィルターに一致	ि नै	バッグ(D)」	をクリック	「回保管した状	態に戻す(V)	適用(Y)	
?					デバッグ(D)	閉じる	

(7) 書き込み完了後、「再開」ボタンをクリックしてください。



(8) バンク0にあるユーザプログラム1の動作確認が終了後、「中断」ボタンをクリックしてください。



次にユーザプログラム2の動作を確認するため、バンク1を起動バンクに切り替える手順を説明します。 (9) 再度、「実行(R)」→「デバッグの構成(B)」の順にクリックしてください。

s Views	実行	テ( <b>R</b> ) ウィンドウ(W)	) ヘルプ(H)	
1 Hardv	Q。 格	実行(R) デバッグ(D)	「実行(R)タブ」をク	フリック
0	0	実行履歴(T) 実行(S) 実行構成(№		>
	*	デバッグ度因 デバッグ(G) デバッグの構成(B)	バックの構成(B)」る 	£クリック →
	<b>Q</b>	外部ツール(E)		>



- (10)「Debugger」タブ内の「Connection Settings」タブをクリックしてください。
- (11)「起動バンク」を「バンク 1」に変更してください。
- (12)「デバッグ(D)」をクリックしてください。

5DB Settings Connection Settings デバッグ・ツール設定	ection Settings」タフをクリック	
動作周波数 [MHz]	120.000	^
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	する はい	$\sim$
৵ ターゲット・ポードとの接続		
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	$\sim$
Fine ボーレート[Mbps]	1.50	~
ホット・プラグ	いいえ	$\sim$
✓ 電源		
エミュレーターから電源を供給する (MAX 200mA)	いいえ	$\sim$
供給電圧 (V)	3.3	$\sim$
✓ 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	$\sim$
起動バンクを変更する	はい	~
起動バンク	パンク1	~
~通「起動バンク」を「バンク 1」に変更 /	デバッグ・モード	× .

(13) 書き込み完了後、「再開」ボタンをクリックしてください。

実行(R)	ウィンドウ(W) ヘルプ(H)				
- 🗂	🔚 🐚 😽 🕶 🚳 👻 📭 💷 🛤 🎿 👁 🕼 🖬 🤫 🍪 🐇 🐐 🗣 💊 🕶 🍇				
	「再開」ボタンをクリック				
are1.c	implication in the set of the				
	<pre>extern void vsta_knl(void); #endif/* BSP_CFG_RTOS_USED */</pre>				
	<pre>     Private global variables and functions     /* Power-on reset function declaration */     R_BSP_POR_FUNCTION(R_BSP_STARTUP_FUNCTION); </pre>				
	<pre>/* Main program function declaration */ @ #if (BSP_CFG_RTOS_USED == 0)    (BSP_CFG_RTOS_USED == 5) /* Non-OS or Azu</pre>				



(14) バンク1にあるユーザプログラム2が動作していることを確認してください。

(15)「終了」ボタンをクリックするとデバッグが終了します。

実行(R)	ウィンドウ(W) ヘルプ(H)	è		
📬 🕶 🛛	3 🕼   🕲 • 🔦 •	📸 i 🗙 i 🕪 🔟 🔳 🖡	◆   彩   定 戸 1   3. の 広 1	• 💁 •   001 • 🕸
	0		「終了」ボタンをクリック	
are1.c $\times$	firmware1.scfg	c resetprg.c		
	<pre># FILE #include "r_sm #include "r_fl #include "Confi </pre>	: firmware1.c[ c_entry.h" ash_rx_if.h" ig_PORT.h"		
	void main(void	);		
f00706	<pre>void main(void</pre>	)		



3.4 ユーザプログラムの mot ファイルを出力する方法

e<sup>2</sup> studio 上で mot ファイルを出力する方法を以下に示します。

オフセットの有無によって方法が異なります。通常の mot ファイルの出力は「3.4.1 mot ファイルの出力 手順」を、オフセットした mot ファイルの出力は「3.4.2 オフセットした mot ファイルの出力手順」を参照 してください。

3.4.1 mot ファイルの出力手順

通常の mot ファイルを出力する手順を以下に示します。

使用するコンパイラによって方法が異なります。「Renesas CC-RX」を使用している場合は 「3.4.1.1 Renesas Electronics C/C++ Compiler Package for RX Family を使用する場合」を、「GCC for Renesas RX」を使用している場合は「3.4.1.2 GCC for Renesas RX を使用する場合」を参照してください。

3.4.1.1 Renesas Electronics C/C++ Compiler Package for RX Family を使用する場合

(1) e<sup>2</sup> studio を起動してください。

(2)「プロジェクト・エクスプローラ」上で出力するプロジェクトをクリックしてください。



(3)「プロジェクト(P)」→「プロパティ(P)」の順にクリックしてください。





(4)「C/C++ ビルド」の「>」をクリックしてください。

フィルタ入力 「>」をクリック (AUK)	
<u>フィルタ入ヵ</u> <u>&gt; リソース</u> [>」をクリック	
> 1/2 [> 27 1 97	
> C/C++ ビルド > C/C++ 一般 Renesas QE ビルダー プロジェクト・ネーチャー プロジェクト参照 実行/デバッグ設定 9 イブ(Y): タイブ(Y): 最終変更日時(1 テキスト・ファイ)I ○ コンテナーが ④ その他(O): □ 派生リソーフ 新規テキスト・フ ③ コンテナー (V ○ その他(H):	/firmv プロジュ C:¥wo M): 2023年 レ・エンコード( 5継承(I) (U UTF-8 のエンコード ファイルの行足 Vindows) が Windows

- (5)「設定」をクリックしてください。
- (6)「ツール設定」タブをクリックしてください。
- (7)「Converter」から「出力」をクリックしてください。
- (8)「モトローラS形式ファイルを出力する (-form=stype)」にチェックを入れてください。
- (9)「適用して閉じる」をクリックしてください。





(10)「プロジェクト(P)」→「プロジェクトのビルド(B)」の順にクリックしてください。



(11) ビルド完了後、プロジェクト内の「HardwareDebug」フォルダ内に mot ファイルが出力されます。





3.4.1.2 GCC for Renesas RX を使用する場合

(1) e<sup>2</sup> studio を起動してください。

(2)「プロジェクト・エクスプローラ」上で設定を変更するプロジェクトをクリックしてください。

e2_studio - e <sup>2</sup> studio	
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 Views 実行
🔦 🎋 🔳 🎋 デバッグ(B)	<ul> <li>Firmware2 HardwareD</li> </ul>
<u> プロジェクト・エクスプロ−ラ− ×</u>	🖻 😫 🍸 🕴 🗖 🖬
> 🥵 firmware1 [HardwareDebug]	
> 😸 firmware2	
	プロジェクトをクリック

(3)「プロジェクト(P)」→「プロパティ(P)」の順にクリックしてください。



(4)「C/C++ ビルド」の「>」をクリックし、プルダウンメニューを展開してください。





- (5)「設定」をクリックしてください。
- (6)「ツール設定」タブをクリックしてください。
- (7)「Objcopy」から「General」をクリックしてください。
- (8)「OutFormat」を「Motorola S-record (srec)」に変更してください。
- (9)「適用して閉じる」をクリックしてください。



(10)「プロジェクト(P)」→「プロジェクトのビルド(B)」の順にクリックしてください。



(11) ビルド完了後、プロジェクト内の「HardwareDebug」フォルダ内に mot ファイルが出力されます。





3.4.2 オフセットした mot ファイルの出力手順

- (1) e<sup>2</sup> studio を起動してください。
- (2)「実行(R)」→「デバッグの構成(B)」の順にクリックしてください。

Hardy	Q,	実行(R)		
8 =	樯	デバッグ(D)	「美行(R)」 ダブをク	リック
0		実行履歴(T)		>
	0	実行(S)		>
		実行構成()**	バッグの構成(B)」を	ミクリック
	な	デ <i>ド</i> ッグ(G)		>
		デバッグの構成(B)		
	0	外部ツール(F)		>

- (3)「Debugger」タブ内の「Connection Settings」タブをクリックしてください。
- (4)「起動バンク」を「バンク 0」に変更してください。

デバッグ構成 構成の作成、管理、および実行	r	Debugger」タブをクリック	×
1 6 9 B X B 7 •	名前(N): firmware1 Hardwa eDebug	Connection Settings	タブをクリ
フィルタ入力	● メイン 莎 Debugger ▶ Startup □ 共通(C	x-x	
<ul> <li>C C/C++ アプリケーション</li> <li>C C/C++ リモート・アプリケーション</li> <li>■ EASE Script</li> <li>GDB Simulator Debugging (RH850)</li> </ul>	Debug hardware: E2 Lite (RX) イ Targ GDB Settings Connection Settings デパッグ・ツ	pet Device: R5F5671E_DUAL …	
GDB ハードウェア・デバッギング	エミュレーター	(Auto)	
Java アプリケーション	接続タイプ	JTag	~
🗾 Java アプレット	JTag クロック周波数[MHz]	6.00	~
Renesas GDB Hardware Debugging	Fine ボーレート[Mbps]	1.50	~
c firmware1 HardwareDebug	ホット・プラグ	いいえ	~
firmware2 HardwareDebug	~ 電源		
💽 Renesas Simulator Debugging (RX, RL78)	エミュレーターから電源を供給する (MAX 200r	nA) いいえ	~
リモート Java アプリケーション	供給電圧 (V)	3.3	×.
🖏 起動グループ	✓ 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	~
	起動パンクを変更する	はい	~
	起動バンク	バンクロ	~
	◇ 通信モード		
Г±		赤田 デバッグ・モード	~
「定	『劉ハンク』を「ハンク 0」に	多史	~
	> フラッシュ		
	ID J-F	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	FFFFF

(5) FFF0 0000h~FFFF FFFFh に配置したプログラムを、オフセットを付けて FFE0 0000h~FFEF FFFFh にダウンロードします。詳細は「3.3 起動バンク切り替え機能のデバッグ方法」の手順(3)~(6)を実施し てください。

	新規ウィンドウ(N)				
.c	Iテ₁タ- >	X			
a00 ⊝	外観 >	STA	RTUP_FUNCTION)		
	ビューの表示(V) >	Ec	C/C++ プロジェクト		
	バースペクティブ(や) >	\$	Terminal		
Θ	ナドゲーション(G)		アウトライン	Alt+シフト+Q,O	
		ź	ーの表示(V)」を	<b>クリック</b> フト+Q,C	
Θ	設定(P)		<b>人マート・ノフワサー</b>		
Θ	<pre>#if defined(GNUC)</pre>	ц,	スマート・マニュアル		
Θ	#if BSP_CFG_USER_STACK_E	2	タスク		
	#endif	۲	ビルド・ターゲット		
	INTERNAL_NOT_USED(is	· 🔁	プロジェクト・エクスプローラー		
	#enalt		プロパティー		
Θ	<pre>#if defined(CCRX)   </pre>	A	検索	Alt+シフト+Q,S	
	/* Tnitializa tha Tn		最適化アシスタント		
<sup>30</sup> 「そ(	の他(o)…」をクリック	7 :	組み込みブラウザー		
Θ	#ifdef BSP MCU EXCLPTION		問題	Alt+シフト+Q,X	
	/* Initialize the Ex		問題の詳細		
017	<pre>R_BSP_SET_EXTB(R_BSP</pre>				

(6)「ウィンドウ(W)」→「ビューの表示(V)」→「その他(o)…」の順にクリックしてください。

(7)「ビューの表示」に「Debugger Console」と入力し、「開く(o)」をクリックしてください。



📴 ビューの表示				×
Debugger Console				×
✓ 🗁 デバング 🛃 Debugger Cor	nsole			
۲Debugger Co	onsole	s r	:入力	,
「開く(o)	」をク	リッ	ック	
	/			
Щ	K(o)		キャンセ	JL



- (8) オフセットを付けて FFE0 0000h~FFEF FFFFh にダウンロードしたプログラムの mot ファイルを出力 するため、以下の GDB コマンドを実行してください。
   mot ファイルは任意のファイル名を入力してください。
  - コマンド: dump srec memory firmware2.mot 0xffe00000 0xffefffff

💷 コンソール 🔝 問題 👒 スマート・ブラウザー 🙀 Debugger Console 🗙
firmware1 HardwareDebug [Renesas GDB Hardware Debugging] rx-elf-gdb -rx-force-isa=v3 -rx-force-double-fpu (7.8.2)
monitor set_io_access_width,RW,1,913b6-913bd,913c6-913cd,913d6-913dd,913e6-913ed,913f6-913f
monitor set_io_access_width,RW,1,91829-9182a,9182a-9182b,9182b-9182c,9182c-9182d,9182d-9182
monitor set_io_access_width,RW,1,91837-91838,91838-91839,91839-9183a,9183a-9183b,9183b-9183
<pre>monitor set_io_access_width,RW,1,a0b00-a0b03,a0b05,a0b0c-a0b0f,a0b17-a0b1f,c1200-c120a,c120</pre>
monitor set_io_access_width,RW,1,c1298-c1299,c1300-c1304,c1324,c1326,c1328,c1380-c1382,c138
monitor set_io_access_width,RW,
monitor set_io_access_width,RWコマンドを人力し、Enter キーを押下 1052-d0053,d005
monitor set_io_access_width_W,_,ucrue ucru,ucrue ucru,ucrue ucru,ucrue uJ161,d0300-d030
monitor_set_io_access_widt#_RV_1.7te010.7te014_7te018,7te0d0,7te0d4
dump srec memory tirmware2.mot 0xtte00000 0xttettttt
C

(9) mot ファイルはプロジェクトフォルダに出力されます。





4. RFPを使ったデュアルモードのプログラム書き込み方法

4.1 出荷状態のマイコンのバンク0に書く方法

「3.1 ユーザプログラム Ver1.00 を開発する場合」の手順で生成した、バンク 0 に書くプログラムとオプション設定メモリの設定が記述された 1 つの mot ファイルを準備します。RFP を使用して以下の手順で書き込むと、マイコン起動時にデュアルモードになり、バンク 0 に書き込んだプログラムが動作します。

(1) RFP を起動してください。

(2)「ファイル(F)」タブをクリックしてください。

(3)「新しいプロジェクトを作成(N)」をクリックし、リニアモードのプロジェクトを作成してください。

Renesas Flash Programmer V3.11.02	ー「ファイル(F)」タブをクリック	- 🗆	$\times$
ファイル(F) <u>ヘルプ(H)</u>			
新しいプロジェクトを作成(N)			
<b>プロジェクトを開く(O)</b> プロジェクトを保存(S)	「新しいプロジェクトを作	₣成(N)」をクリック	
イメージファイルを保存(I) ファイルチェックサム(C)			
ファイルパスワード設定(P)			
終了(X)		参照(B)	
フラッシュ操作			
スタ	—		

- (4)「プロジェクト名(N)」には任意のプロジェクト名を入力してください。
- (5) その他設定は、システムに応じて変更してください。
- (6)「接続(O)」をクリックしてください。

■ 新しいプロジェクトの作成 -	-		×
プロジェクト情報			
マイクロコントローラ(M): RX67x 任意のプロミ	ジェク	ト名を	入力
プロジェクト名(N): rx671_linear			
作成場所(F): C:¥workspace¥Renesas Flash Programmer¥V3.11	参照	ξ(B)	
通信 ツール(T): E2 emulator Lite 〜 インタフェース(D: 2 wire UART 〜 ツール詳細(D) 番号: 自動選択 電源: 供給しない			
「接続(O)」をクリック 	4	キャンセノ	۱(C)



(7)「ID コード(I)」には設定しているセキュリティ ID を入力してください。

(8)「OK」をクリックしてください。

💋 認証	セキュリティ ID を入力 🚽 📃 🛛 🗙
認証コード	
ID⊐' *(I):	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
└────── <mark>「OK</mark>	、」をクリック OK キャンセル(C)

(9)「操作設定」タブをクリックしてください。

(10)「コマンド」は「書き込み(P)」と「ベリファイ(V)」にチェックを入れてください。

Renesas Flash Programmer V3.11.02	– 🗆 X
ファイル(F) ターゲットデバイ「操作設定」タブをク	<b>パリック</b>
操作 操作設定 プロック設定 接続設定 ユニークコ	<b>ド</b>
<ul> <li>コマンド</li> <li>□ 消去(E)</li> <li>□ 書き込み(P)</li> <li>□ パリファイ(V)</li> <li>□ チェックサム(S)</li> </ul>	消去オブション(0) ブロック選択消去 書き込みとベリファイオブション 「ベリファイ(V)」にチェックを入れる デバイスからリードしてベリファイ
0xFF補完 □ Code Flash / User Boot □ Data Flash	チェックサム計算方式(M) 32bit 加算方式 ン エラー設定 ン デバイス範囲外エラーを有効にする(R)

(11)「操作」タブをクリックしてください。

- (12)「エンディアン(E)」はユーザプログラムに対応したエンディアンを選択してください。
- (13)「プログラムファイル」に書き込むユーザプログラムを選択してください。
- (14)「スタート(S)」をクリックしてください。
- (15)「操作が成功しました。」と表示されると書き込み完了です。

Renesas Flash Proceedings 11 02 「操作」タブをクリック ファイル(F) ダーゲーレーロンド・シング・リック	- 🗆 X
操作 操作設定 ブロック設定 接続設定 ユニークコード	
プロジェクト情報 現在のプロジェクト: rx671_linear.rpj マイクロコ <mark>「プログラムファイル」にユーザプログラムを選択</mark> エンデ	デ <b>ィアンを選択</b> ィアン(E): リトル
プログラムファイル	4 m (n)
C:#workspace#e2_studio#firmware1#HardwareDebug#firmware1.mot CRC-32	· 9A22FF33
フラッジュ操作 「スタート(S)」をクリック	
消去 >> 書き込み >> ベリファイ	
スタート(S)	正常終了
[Config Area] 0×FE7F5D00 - 0×FE7F5D1F サイズ:32 [Config Area] 0×FE7F5D40 - 0×FE7F5D7F サイズ:64	^
[Config Area] 0xFE7F5D00 - 0xFE7F5D1F       サイズ: 32         [Config Area] 0xFE7F5D40 - 0xFE7F5D7F       サイズ: 64         ベリファイを実行します。       [Config Area] 0xFE7F5D00 - 0xFE7F5D1F       サイズ: 32         [Config Area] 0xFE7F5D40 - 0xFE7F5D1F       サイズ: 64         ツールから切断します。       「操作が成功しました。」と表示される	^
[Config Area] 0xFE7F5D00 - 0xFE7F5D1F サイズ:32 [Config Area] 0xFE7F5D40 - 0xFE7F5D7F サイズ:64 ベリファイを実行します。 [Config Area] 0xFE7F5D00 - 0xFE7F5D1F サイズ:32 [Config Area] 0xFE7F5D40 - 0xFE7F5D7F サイズ:64 ツールから切断します。 <b>操作が成功しました。」と表示される</b>	



#### 4.2 バンク1に書く方法

「4.1 出荷状態のマイコンのバンク0に書く方法」でバンク0にユーザプログラムを書いた後、バンク1に 書く方法を示します。

「3.4.2 オフセットした mot ファイルの出力手順」で生成した、バンク 1 に書く mot ファイル(オプション設定メモリの設定なし)を準備してください。

(1) RFP を起動してください。

- (2)「ファイル(F)」タブをクリックしてください。
- (3)「新しいプロジェクトを作成(N)」をクリックし、デュアルモードのプロジェクトを作成してください。

🌠 Renesas Flash Programmer V3.11.02	ー「ファイル(F)」タブをクリック ー 🗆 🗙
ファイル(F) <u>ヘルプ(H)</u>	
新しいプロジェクトを作成(N)	
<b>プロジェクトを開く(O)</b> プロジェクトを保存(S)	「新しいプロジェクトを作成(N)」をクリック
イメージファイルを保存(I) ファイルチェックサム(C)	
ファイルパスワード設定(P)	
終了(X)	参照(B)
フラッシュ操作	
スタ	-  (S)

- (4)「プロジェクト名(N)」には任意のプロジェクト名を入力してください。
- (5) その他設定は、システムに応じて変更してください。
- (6)「接続(O)」をクリックしてください。

🌠 新しいプロジェクトの作成	_		×
プロジェクト情報			
マイクロコントローラ(M): RX67x 任意の	プロジェ	ェクト名	を入力
プロジェクト名(N): rx671_dual			
作成場所(F): C:¥workspace¥Renesas Flash Programmer¥V3.	11 :	参照(B)	
通信			
ツール(T): E2 emulator Lite 🗸 インタフェース(I): 2 wire UART 🧹			
ツール詳細(D) 番号: 自動選択 電源: 供給しない			
「接続(O)」をクリック	_		
	(0)	キャンセ	л(с)



(7)「ID コード」には設定しているセキュリティ ID を入力してください。

(8)「OK」をクリックしてください。

💋 認証	セキュリティ ID を入力 🚽 📃 🛛 🗙
認証コード	
ID⊐' *(I):	FFFFFFFFFFFFFFFFFFFFFFFFFFFFF
└── 自動認証	」をクリック OK キャンセル(C)

(9)「操作設定」タブをクリックしてください。

(10)「コマンド」は「書き込み(P)」と「ベリファイ(V)」にチェックを入れてください。

Renesas Flash Programmer V3.11.02	- 🗆 X
ファイル(F) ターゲットデバイ「操作設定」タブをク	フリック
操作 操作設定 クロック設定 接続設定 ユニークコ	コード
אעדב	消去オブション(0)
□ 消去(E)	ブロック選択消去 ~
☑ 書き込み(P)	書き込みとベリファイオプション
✓ パリファイ(V) 「書き込み(P)」と	「ベリファイ(V)」にチェックを入れる
<ul> <li>チェックサム(S)</li> </ul>	デバイスからリードしてベリファイ 🗸
	チェックサム計算方式(M)
	32bit 加算方式 ~
0xFF補完	
Code Flash / User Boot	エラー設定
Data Flash	✓ デバイス範囲外エラーを有効にする(R)

(11)「操作」タブをクリックしてください。

- (12)「エンディアン(E)」はユーザプログラムに対応したエンディアンを選択してください。
- (13)「プログラムファイル」に書き込むユーザプログラムを選択してください。
- (14)「スタート(S)」をクリックしてください。
- (15)「操作が成功しました。」と表示されると書き込み完了です。

Renesas Flash F 「操作」タブをクリック ファイル(F) ケリアノハロス(ロ)	_		×
操作 操作設定 ブロック設定 接続設定 ユニークコード			
プロジェクト情報 ユーザプログラムに対応したエンデ 現在のプロジェクト: rx671_dualrpj	ィアンを選択	र	
<sup>マイクロゴ</sup> 「プログラムファイル」にユーザプログラムを選択 <sup>エンディア</sup>	"ン(E): リトル	~	]
プログラムファイル			
C:¥workspace¥firmware2.mot	Ť	:照(B)	
CRC-32:2 フラッシュ操作 書き込み >> ベリファイ	7D79BF6		_
	_		- 1
スタート(S)	正常	終了	
スタート(S) 書き込みを実行します。 [Code Flash 1] 0xFFE00000 - 0xFFEFFFFF サイズ: 1.0 M ベリファイを実行します。 [Code Flash 1] 0xFFE00000 - 0xFFEFFFFF サイズ: 1.0 M ツールから切断します。 操作が成功しました。」と表示される	正常	終了	*



# 5. 注意事項

# 5.1 オプション設定メモリの設定

オプション設定メモリの設定をソースコード上で記述可能ですが、ver.1.01以降のユーザプログラムでは BANKSEL.BANKSWP[2:0]ビットは設定しないでください。セルフプログラミングでのみ切り替えを行って ください。



# 6. 参考ドキュメント

ユーザーズマニュアル:ハードウェア RX671グループ ユーザーズマニュアル ハードウェア編 (R01UH0899JJ) (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。)

アプリケーションノート

RXファミリ ボードサポートパッケージモジュール Firmware Integration Technology (R01AN1685JJ) (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。)

RXファミリ フラッシュモジュール Firmware Integration Technology (R01AN2184JJ) (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。)

RXファミリ ファームウェアアップデートモジュール Firmware Integration Technology (R01AN5824JJ) (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。)

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。)



# 改訂記録

		改訂内容		
Rev.	発行日	ページ	ポイント	
1.00	Jun.30.23	—	初版発行	



#### 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<sub>IL</sub>(Max.)から V<sub>IH</sub>(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<sub>IL</sub>(Max.)からV<sub>IH</sub> (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。

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

8. 製品間の相違について

型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違うと、フラッ シュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ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

- 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。
- 当社製品または本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許 権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うもので はありません。
- 3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
- 4. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要と なる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
- 5. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改 変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
- 6. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図 しております。

標準水準: コンピュータ、OA機器、通信機器、計測機器、AV機器、家電、工作機械、パーソナル機器、産業用ロボット等 高品質水準:輸送機器(自動車、電車、船舶等)、交通制御(信号)、大規模通信機器、金融端末基幹システム、各種安全制御装置等 当社製品は、データシート等により高信頼性、Harsh environment向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある 機器・システム(生命維持装置、人体に埋め込み使用するもの等)、もしくは多大な物的損害を発生させるおそれのある機器・システム(宇宙機器 と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等)に使用されることを意図しておらず、これらの 用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責 任を負いません。

- 7. あらゆる半導体製品は、外部攻撃からの安全性を100%保証されているわけではありません。当社ハードウェア/ソフトウェア製品にはセキュリ ティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害(当社製品または当社製品が使用されてい るシステムに対する不正アクセス・不正使用を含みますが、これに限りません。)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、脆弱性問題に起因しまたはこれに関連して生じた損害に ついて、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア/ソフトウェア製品について、商品 性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
- 8. 当社製品をご使用の際は、最新の製品情報(データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等)をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
- 9. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする 場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment向け製品と定義しているものを除き、耐放射線設計を 行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客 様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を 行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行って ください。
- 10. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用 を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことに より生じた損害に関して、当社は、一切その責任を負いません。
- 11. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
- 12. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたしま す。
- 13. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
- 14. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。
- 注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的 に支配する会社をいいます。
- 注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。

(Rev.5.0-1 2020.10)

#### 本社所在地

#### 〒135—0061 東京都江東区豊洲3—2—24 (豊洲フォレシア) www.renesas.com

# お問合せ窓口

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

www.renesas.com/contact/

# 商標について

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