お客様各位

カタログ等資料中の旧社名の扱いについて

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願い申し上げます。

ルネサスエレクトロニクス ホームページ（http://www.renesas.com）

2010年4月1日
ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社（http://www.renesas.com）

【問い合わせ先】http://japan.renesas.com/inquiry

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。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。
3. 当社製品を改造、改変、複製等しないでください。
4. 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。お客様の機器の設計において、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には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、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に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したものですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。

   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

   高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）

   特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障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おりますが、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、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害など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、機器またはシステムとしての出荷保証をお願いいたします。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、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、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。お客様が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. 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固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

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その総株主の議決権の過半数を直接または間接に保有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。

H8S ファミリ

ユーザブートモードでの内蔵フラッシュメモリ書き換え例 「SCI (クロック同期式)」

要旨

本アプリケーションノートは，H8S/2556,2552,2506 をユーザブートモードで動作させ，シリアルコミュニケーションインタフェース (以下 SCI) のクロック同期式を使用した内蔵フラッシュメモリ (ユーザマット) 書き換え動作例をまとめたものです。

動作確認デバイス

H8S/2500 シリーズ H8S/2556

目次

1. 仕様

本アプリケーションノートでは，H8S/2556 をユーザブートモードで起動し，フラッシュメモリの EB10～12 の 3 ブロックを消去後，書き込みデータ要求信号を送信し，受信した書き込みデータを EB10～12 の 3 ブロックに書き込みます。インタフェースはシリアルコミュニケーションインタフェース (以下 SCI) を使用し，クロック同期式とします。なお，H8S/2556 は SCI2 を使用します。

図 1 に SCI (クロック同期式) を使用したオンボード書き換え構成図を示します。

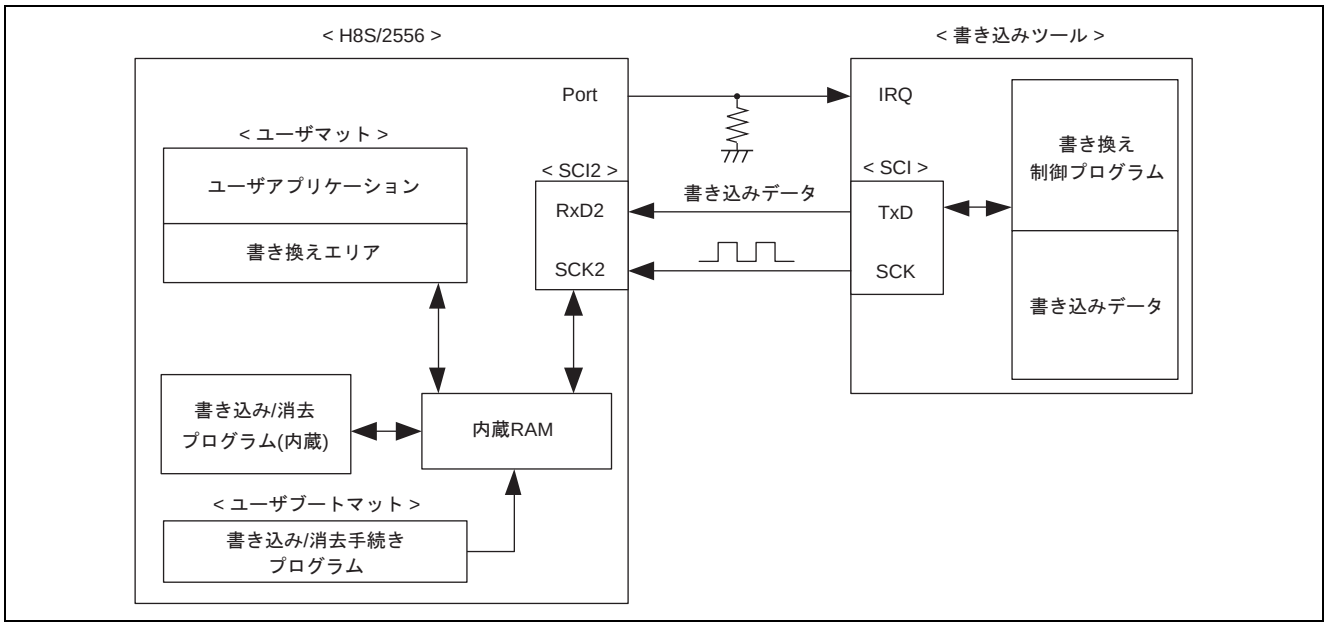


図 1　SCI (クロック同期式) を使用したオンボード書き換え構成図

1.1　動作モード

ユーザブートモードの端子設定を以下に示します。

表 1　端子設定

端子	レベル
$\overline{\text{RES}}$	1
MD0	1
MD1	0
MD2	0

1.2　本アプリケーション例のソフトウェア開発環境

本アプリケーション例のソフトウェア開発には，HEW3 (Ver3.01.06.001)，ユーザブートモードでのソフトウェア書き込みに FDT2.0 を使用しています。

1.3 インタフェース仕様

SCI インタフェース仕様を以下に示します。

表 2 インタフェース仕様

	H8S/2556	書き込みツール
使用チャネル	チャネル 2 (SCI2) 受信のみ	送信のみ
クロック (SCK)	入力	出力
コミュニケーションモード	クロック同期式	
ボーレート	1Mbps	

1.4 制御仕様

制御仕様を以下に示します。

表 3 制御仕様

制御	H8S/2556	書き込みツール
書き込みデータ要求信号	ポート PJ0 (53Pin：出力)	IRQ (立ち下がりエッジ)

1.5 制御信号動作説明

本アプリケーションノートでは，書き込みデータ要求信号にポートを使用します (フラッシュメモリ書き込み/消去の処理中の割り込み禁止)。

フラッシュメモリ書き換えを終了するまでの制御動作を図 2 に示します。

なお，本アプリケーション例ではエラー処理を定義していません。本アプリケーション例で使用する書き込みツールは，書き換え開始信号を送信後，H8S/2556 から一定の期間書き込みデータ要求信号がない場合には書き込みが正常に終了したと判断する仕様となっています。

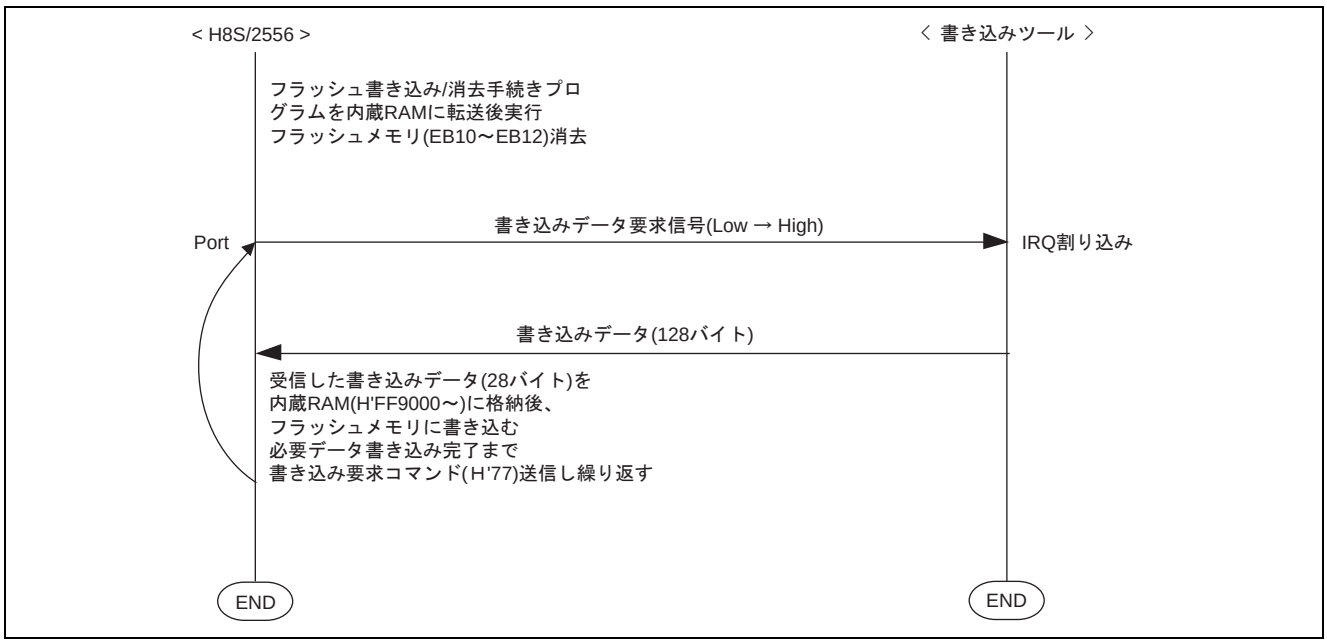
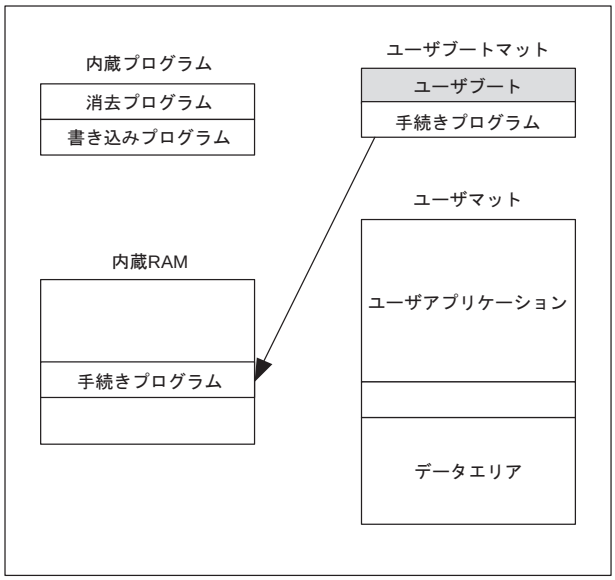


図 2 制御動作

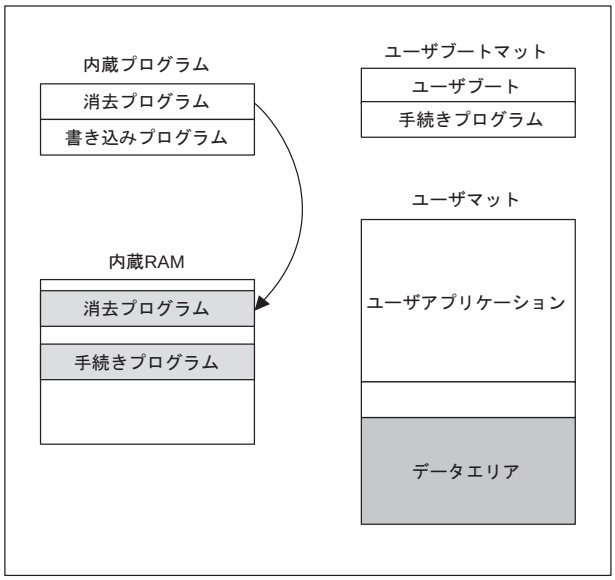
2. ソフトウェア動作説明

本アプリケーション例のソフトウェア動作説明を以下に示します。

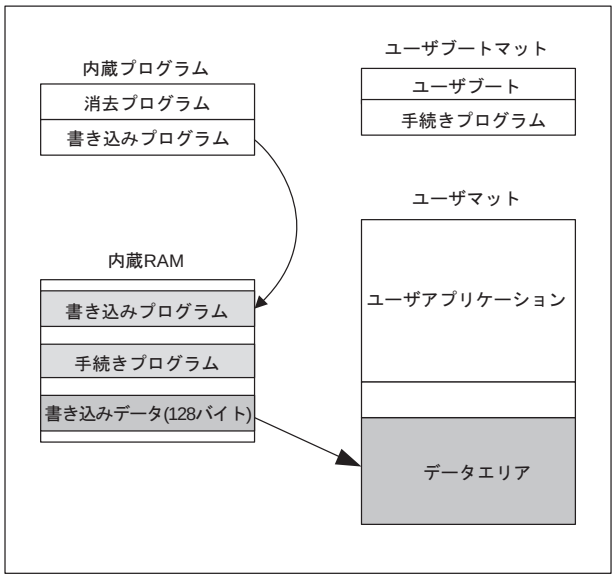
(1) 起動後、ユーザブートプログラムによりフラッシュ消去/書き込み手続きプログラムを内蔵RAMに転送し、内蔵RAM上で手続きプログラムを実行。



(2) 手続きプログラムは内蔵RAM上で、内蔵プログラム (消去プログラム) 内蔵RAM上にダウンロードし、指定したブロックエリアのフラッシュメモリを消去します。



(3) 手続きプログラムは内蔵RAM上で、内蔵プログラム (書き込みプログラム) を内蔵RAMにダウンロードし、書き込みツールから受信した書き込みデータをいったん内蔵RAMに格納し、指定した先頭アドレスから128バイトごとにフラッシュメモリに書き込みます。



3. レジスタ説明

フラッシュメモリをコントロールするレジスタ/パラメータを以下に示します。

RAMER 以外のフラッシュメモリをコントロールするレジスタにアクセスするためには，内蔵フラッシュメモリが有効なモードで SYSCR2 の FLSHE ビットを 1 にセットする必要があります。ただし，FLSHE = 1 のときは一部の TPU 制御レジスタ (H'FFFE80～H'FFFEB1) をアクセスできません。TPU のレジスタをアクセスする際は必ず FLSHE ビットをクリアしてください。

3.1 書き込み/消去インタフェースレジスタ

書き込み/消去インタフェースレジスタについて説明します。すべて 8 ビットのレジスタでバイトアクセスのみ可能です。FCCS レジスタの FLER ビットを除き，これらのレジスタはパワーオンリセットとハードウェアスタンバイモード/ソフトウェアスタンバイモード/ウォッチモードで初期化されます。FLER ビットは，ソフトウェアスタンバイモード，ウォッチモードでは初期化されません。

3.1.1 フラッシュコードコントロール・ステータスレジスタ (FCCS) 初期値：H'80

FCCS は，フラッシュメモリの書き込み/消去実行中のエラー発生のモニタ，および内蔵プログラムのダウンロードを要求するビットから構成されています。

ビット	ビット名	R/W	説明
7	–	R	リザーブビット リードすると常に 1 が読み出されます。ライトも常に 1 にしてください。
6,5	–	R	リザーブビット リードすると常に 0 が読み出されます。ライトも常に 0 にしてください。
4	FLER	R	フラッシュメモリエラー フラッシュメモリへの書き込み/消去実行中にエラーが発生したことを示すビットです。FLER = 1 にセットさせると，フラッシュメモリはエラープロテクト状態に遷移します。パワーオンリセットまたはハードウェアスタンバイモード遷移で初期化されます。なお，FLER = 1 になった場合は，フラッシュメモリ内部に高電圧が印加されていますので，フラッシュメモリへのダメージを低減するために，通常より長い 100μs のリセット入力期間の後にリセットリリースしてください。 0：フラッシュメモリは正常に動作しています。 フラッシュメモリへの書き込み/消去プロテクト (エラープロテクト) は無効 ［クリア条件］パワーオンリセットまたはハードウェアスタンバイモードのとき 1：フラッシュメモリへの書き込み/消去中にエラーが発生したことを示します。フラッシュメモリへの書き込み/消去プロテクト (エラープロテクト) が有効
3~1	–	R	リザーブビット リードすると常に 0 が読み出されます。ライトも常に 0 にしてください。

(次頁へ続く)

ビット	ビット名	R/W	説明
0	SC0	(R)/W	ソースプログラムコピーオペレーション 内蔵されている書き込み/消去プログラムを内蔵RAMにダウンロードする要求ビットです。本ビットに 1 を書き込むと，FPCS/FECS レジスタで選択した内蔵プログラムが，FTDAR レジスタで指定された内蔵 RAM の領域に自動的にダウンロードされます。本ビットに 1 を書き込むためには，RAM エミュレーション状態の解除，FKEY レジスタへの H'A5 の書き込み，および内蔵 RAM 上での実行が必要です。 本ビットに 1 を書き込んだ直後には，4 個の NOP 命令を必ず実行するようにしてください。なお，ダウンロード完了時点では本ビットは 0 クリアされているため，本ビットの 1 状態を読み出すことはできません。 0：内蔵されている書き込み/消去プログラムの内蔵 RAM へのダウンロードは行いません。 [クリア条件] ダウンロードが完了するとクリアされます。 1：内蔵されている書き込み/消去プログラムの内蔵 RAM へのダウンロードリクエストを発生します。 [セット条件] 以下の条件がすべて満足されている状態で，1 を書き込んだとき • (1) FKEY レジスタに H'A5 が書かれていること • (2) 内蔵 RAM 上で実行中であること • (3) RAM エミュレーションモードでないこと (RAMER の RAMS = 0 であること)

3.1.2 フラッシュプログラムコードセレクトレジスタ (FPCS) 初期値：H'00

FPCS は，ダウンロードする書き込み関係の内蔵プログラムを選択するレジスタです。

ビット	ビット名	R/W	説明
7~1	–	R	リザーブビット リードすると常に 0 が読み出されます。ライトも常に 0 にしてください。
0	PPVS	R/W	プログラムパルスベリファイ 書き込みプログラムを選択します。 0：内蔵の書き込みプログラムを選択しません。 [クリア条件] 転送が終了するとクリアされます。 1：内蔵の書き込みプログラムを選択します。

3.1.3 フラッシュイレースコードセレクトレジスタ (FECS) 初期値：H'00

FECS は，消去関係の内蔵プログラムのダウンロードを選択するレジスタです。

ビット	ビット名	R/W	説明
7~1	–	R	リザーブビット リードすると常に 0 が読み出されます。ライトも常に 0 にしてください。
0	EPVB	R/W	イレースパルスベリファイブロック 消去プログラムを選択します。 0：内蔵消去プログラムを選択しません。 [クリア条件] 転送が終了するとクリアされます。 1：内蔵消去プログラムを選択します。

3.1.4 フラッシュキーコードレジスタ (FKEY) 初期値：H’00

FKEY は，内蔵プログラムのダウンロードとフラッシュメモリの書き込み/消去を許可するソフトウェアプロテクトのレジスタです。内蔵プログラムのダウンロード実施のための SCO ビットへの 1 書き込み前，およびダウンロードした書き込み/消去プログラム実行前に，キーコードを書き込まないとそれぞれの処理が実行できません。

ビット	ビット名	R/W	説明
7	K7	R/W	キーコード H'A5 を書き込んだ場合にのみ，SCO ビットの書き込みが有効になります。H'A5 以外の値が FKEY レジスタに書かれている場合，SCO ビットに 1 を書き込むことができないため，内蔵 RAM へのダウンロードができません。H'5A を書き込んだ場合にのみ，書き込み/消去が可能になります。内蔵の書き込み/消去プログラムを実行しても，H'A5 以外の値が FKEY レジスタに書かれている場合はフラッシュメモリの書き込み/消去はできません。 H'A5：SCO ビットの書き込みを許可します。(H'A5 以外では SCO ビットのセットはできません) H'5A：書き込み/消去を許可します。(H'5A 以外ではソフトプロテクト状態) H'00：初期値
6	K6	R/W	
5	K5	R/W	
4	K4	R/W	
3	K3	R/W	
2	K2	R/W	
1	K1	R/W	
0	K0	R/W	

3.1.5 フラッシュマットセレクトレジスタ (FMATS) 初期値：H’AA *1

FMATS は，ユーザマット/ユーザブートマットのどちらを選択するかを指定するレジスタです。

ビット	ビット名	R/W	説明
7	MS7	R/W	マットセレクト H'AA 以外の場合はユーザマット選択状態，H'AA が書かれている状態はユーザブートマット選択状態です。FMATS に値を書き込むことによりマット切り換えが発生します。 H'AA：ユーザブートマットを選択します。 (H'AA 以外ではユーザマット選択状態となります) ユーザブートモードで立ち上がった場合の初期値です。 H'00：ユーザブートモード以外で立ち上がった場合の初期値です。 (ユーザマット選択状態です) [書き込み可能条件] 内蔵 RAM 上での実効状態であること
6	MS6	R/W	
5	MS5	R/W	
4	MS4	R/W	
3	MS3	R/W	
2	MS2	R/W	
1	MS1	R/W	
0	MS0	R/W	

【注】 1. ユーザブートモードのときは H’AA になります。それ以外のときは H’00 となります。

3.1.6 フラッシュトランスファデスティネーションアドレスレジスタ (FTDAR) 初期値：H'00

FTDAR は，内蔵プログラムのダウンロード先の内蔵 RAM 上のアドレスを指定するレジスタです。FCCS レジスタの SCO ビットに 1 を書き込む前に，本レジスタの設定を行ってください。

ビット	ビット名	R/W	説明
7	TDER	R/W	トランスファデスティネーションアドレス設定エラー TDA6～TDA0 ビットで指定するダウンロード先頭アドレス指定にエラーがあった場合，1 がセットされます。アドレス指定のエラー判定は，FCCS レジスタの SCO ビットを 1 にして，ダウンロード処理が実行されたときに，TDA6～TDA0 の値が H'00～H'07 の範囲にあるかどうかを判定します。SCO ビットを 1 に設定する前に，本ビットの値を 0 にすることも含めて，FTDAR の値を H'00～H'07 の範囲に設定してください。 0：TDA6～TDA0 の設定は，正常値 1：TDER,TDA6～TDA0 の設定値が H'08～H'FF で，ダウンロードは中断したことを示します。
6 5 4 3 2 1 0	TDA6 TDA5 TDA4 TDA3 TDA2 TDA1 TDA0	R/W R/W R/W R/W R/W R/W R/W	トランスファデスティネーションアドレス ダウンロード先頭アドレスを指定します。設定可能な値は H'00～H'07 で，4k バイト単位で内蔵 RAM 上のダウンロード先頭アドレスを指定できます。 H'00：ダウンロード先頭アドレスを H'FF9000 に設定 H'01：ダウンロード先頭アドレスを H'FFA000 に設定 H'02：ダウンロード先頭アドレスを H'FFB000 に設定 H'03：ダウンロード先頭アドレスを H'FFC000 に設定 H'04：ダウンロード先頭アドレスを H'FFD000 に設定 H'05：ダウンロード先頭アドレスを H'FFE000 に設定 H'06：ダウンロード先頭アドレスを H'FF8000 に設定

3.1.7 システムコントロールレジスタ 2 (SYSCR2) 初期値：H'00 *1

SYSCR2 はレジスタアクセスの制御を行います。

ビット	ビット名	R/W	説明
7～4	–	–	リザーブビット 0 をライトしてください。
3	FLSHE	R/W	フラッシュメモリコントロールレジスタイネーブル 0 をライトしてフラッシュメモリの制御レジスタの CPU アクセスを制御します。FLSHE ビットを 1 にセットすると，フラッシュメモリ制御レジスタをリード/ライトすることができます。0 にクリアするとフラッシュメモリの制御レジスタは非選択となります。このときフラッシュメモリ制御レジスタの内容は保持されています。 0：アドレス H'FFFFA4～H'FFFFAF のエリアはフラッシュ制御論理を非選択 1：アドレス H'FFFFA4～H'FFFFAF のエリアはフラッシュ制御論理を選択
2	–	–	リザーブビット 0 をライトしてください。
1,0	–	R/W	リザーブビット 0 をライトしてください。

【注】 1. ビット 7～4, 2 は初期値が不定です。それ以外は 0 です。

3.2 書き込み/消去インタフェースパラメータ

書き込み/消去インタフェースパラメータは，ダウンロードした内蔵プログラムに対して動作周波数，ユーザブランチ先アドレス，書き込みデータの格納場所，書き込み先アドレス，消去ブロックなどの指定および処理結果のやりとりをするものです。このパラメータは，CPU の汎用レジスタ (ER0,ER1) や内蔵 RAM 領域を使用します。パワーオンリセット，ハードウェアスタンバイでの初期値は不定です。

ダウンロード，初期化，内蔵プログラム実行においては，R0L 以外の CPU のレジスタは保存されます。R0L は，処理結果の戻り値が記入されます。R0L 以外のレジスタの保存のためにスタック領域を使用しますので，処理開始においてはスタック領域の確保をしてください (使用スタック領域サイズは，最大 128 バイトです)。

書き込み/消去インタフェースパラメータは，次の 4 項目で使用します。

1. ダウンロード制御
2. 書き込み/消去実行前の初期化実行
3. 書き込み実行
4. 消去実行

それぞれ使用するパラメータは異なります。対応表を以下に示します。

ここで FPFR パラメータは初期化処理，書き込み処理，消去処理において処理結果が戻されますが，処理内容によりビットの意味が異なります。

表 4　使用パラメータと対象モード

パラメータ名	略称	ダウンロード	初期化	書き込み	消去	R/W	初期値	割り当て
ダウンロードパス・フェイルリザルト	DPFR	○				R/W	不定	内蔵 RAM *1
フラッシュパス・フェイルリザルト	FPFR		○	○	○	R/W	不定	CPU の R0L
フラッシュプログラムイレース周波数コントロール	FPEFEQ		○			R/W	不定	CPU の ER0
フラッシュユーザブランチアドレスセットパラメータ	FUBRA		○			R/W	不定	CPU の ER1
フラッシュマルチパーパスアドレスエリア	FMPAR			○		R/W	不定	CPU の ER1
フラッシュマルチパーパスデータデスティネーションエリア	FMPDR			○		R/W	不定	CPU の ER0
フラッシュイレースブロックセレクト	FEBS				○	R/W	不定	CPU の ER0

【注】 1. FTDAR レジスタで指定したダウンロード先の先頭アドレス 1 バイト

3.2.1 ダウンロード制御

内蔵プログラムのダウンロードは，SCO ビットを 1 にセットすることで自動的に行われます。ダウンロードされる内蔵 RAM の領域は FTDAR レジスタで指定した先頭アドレスから 2k バイト分の領域です。ダウンロード制御は書き込み/消去インタフェースレジスタで設定し，戻り値は DPFR パラメータで渡されます。

(1) ダウンロードパス/フェイルパラメータ (DPFR : FTDAR レジスタで指定した内蔵 RAM の先頭アドレス 1 バイト)

ダウンロード結果の戻り値です。ダウンロードが実行できたかどうかは，本パラメータの値で判断します。SCO ビットを 1 にできたかの確認が困難のため，ダウンロード開始前 (SCO ビットを 1 にセットする前) に，FTDAR レジスタで指定した内蔵 RAM の先頭アドレス 1 バイトをダウンロードの戻り値以外 (H'FF など) にして，確実な判断ができるようにしてください。

ビット	ビット名	R/W	説明
7～3	–	–	リザーブビット 値 0 が戻されます。
2	SS	R/W	ソースセレクトエラー検出ビット ダウンロード可能な内蔵プログラムは 1 種類のみ指定できます。2 種類以上の選択を行った場合，選択されていない場合，およびマッピングされていない選択の場合にはエラーとなります。 0：ダウンロードプログラムの選択関係は正常 1：ダウンロードエラー発生 (多重選択または，マッピングされていないプログラム選択)
1	FK	R/W	フラッシュキーレジスタエラー検出ビット FKEY レジスタの値が, H'A5 であるかどうかをチェックした結果を返すビットです。 0：FKEY レジスタの設定は正常 (FKEY = H'A5) 1：FKEY レジスタの設定値エラー (FKEY は，H'A5 以外の値)
0	SF	R/W	サクセス/フェイルビット ダウンロードが正常に終了したかどうかを戻すビットです。内蔵 RAM 上にダウンロードしたプログラムをリードバックし，内蔵 RAM 上に転送できているかの判定結果です。 0：内蔵プログラムのダウンロードは正常終了 (エラー無し) 1：内蔵プログラムのダウンロードが異常終了 (エラーが発生している)

3.2.2 書き込み/消去の初期化

ダウンロードされる書き込み/消去の内蔵プログラムには，初期化プログラムも含まれています。

書き込み/消去では決められた時間幅のパルス印加が必要で，ウェイトループを CPU 命令で構成する方法で規定のパルス幅を作成しています。このため，CPU の動作周波数を設定する必要があります。

これらの設定をダウンロードした書き込み/消去プログラムのパラメータとして設定するのが初期化プログラムです。

(1) フラッシュプログラム/イレース周波数パラメータ (FPEFEQ：CPU の汎用レジスタ ER0)

CPU の動作周波数を設定するパラメータです。

ビット	ビット名	R/W	説明
31～16	F31～F16	R/W	リザーブビット 値 0 を設定してください。
15～0	F15～F0	R/W	周波数設定ビット CPU の動作周波数を設定します。設定値は以下のように算出してください。 • MHz 単位で表現した動作周波数を小数点第 3 位で四捨五入し，小数点第 2 位までとする。 • 100 倍した値を 2 進数に変換し，FPEFEQ パラメータ (汎用レジスタ ER0) に書き込む。 具体例として，CPU の動作周波数が 25.000MHz の場合には，以下のようになります。 25.000 の小数点第 3 位を四捨五入し，25.00。 25.00 × 100 = 2500 を 2 進数変換し，b'0000,1001,1100,0100 (H'09C4) を ER0 に設定。

(2) フラッシュユーザブランチアドレス設定パラメータ (FUBRA：CPU の汎用レジスタ ER1)

ユーザブランチ先のアドレスを設定するパラメータです。書き込み/消去実行時のある決まった処理単位ごとに，設定したユーザプログラムを実行することができます。

ビット	ビット名	R/W	説明
31～0	UA31～ UA16	R/W	ユーザブランチ先アドレス ユーザブランチが必要ない場合には，0 番地 (H'00000000) を設定してください。 ユーザブランチ先は，内蔵プログラムが転送されている RAM 領域以外の RAM 空間または外部バス空間としてください。実行コードのない領域にブランチして暴走しないように注意し，内蔵プログラム領域やスタック領域を破壊しないようにしてください。暴走するとフラッシュメモリの値の保証もできません。 ユーザブランチ先の処理では，内蔵プログラムのダウンロード，初期化，書き込み/消去プログラムを起動しないでください。ユーザブランチ先から復帰時の書き込み/消去の保証ができません。また，すでに準備していた書き込みデータを書き換えないでください。さらに，ユーザブランチ先の処理で書き込み/消去インタフェースレジスタの書き換えや，RAM エミュレーションモードへの遷移を行わないでください。ユーザブランチ処理終了後は，RTS 命令で書き込み/消去プログラムに戻ってください。

(3) フラッシュパス/フェイルパラメータ (FPFR：CPU の汎用レジスタ R0L)
初期化結果の戻り値としての FPFR について説明します。

ビット	ビット名	R/W	説明
7～3	–	–	リザーブビット： 値 0 が戻されます。
2	BR	R/W	ユーザブランチエラー検出ビット 指定されたユーザブランチ先アドレスが，ダウンロードされている書き込み/消去関係プログラムの格納領域以外であるかをチェックした結果を戻します。 0：ユーザブランチアドレス設定は正常値 1：ユーザブランチアドレス設定が異常値
1	FQ	R/W	周波数エラー検出ビット 指定された CPU 動作周波数が，サポートしている動作周波数の範囲にあるかをチェックした結果を戻します。 0：動作周波数の設定は正常値 1：動作周波数の設定が異常値
0	SF	R/W	サクセス/フェイルビット 初期化が正常に終了したかどうかを戻すビットです。 0：初期化は正常終了 (エラー無し) 1：初期化が異常終了 (エラーが発生している)

3.2.3 書き込み実行

フラッシュメモリへの書き込み実行においては，ユーザマット上の書き込み先アドレスと書き込みデータをダウンロードした書き込みプログラムに渡すことが必要です。

1. ユーザマット上の書き込み先の先頭アドレスを汎用レジスタ ER1 に設定してください。このパラメータを FMPAR (フラッシュマルチパーパスアドレスエリアパラメータ) と呼びます。書き込みデータは常に 128 バイト単位ですので, ユーザマット上の書き込み先頭アドレスの境界はアドレスの下位 8 ビット (A7～A0) が，H'00 または H'80 のいずれかとしてください。
2. ユーザマットへの書き込みデータを連続領域に準備してください。書き込みデータは CPU の MOV.B 命令でアクセス可能な連続空間で，内蔵フラッシュメモリ空間以外としてください。書き込みたいデータが 128 バイトに満たない場合でも，ダミーコード (H'FF) を埋め込んで 128 バイトの書き込みデータを準備してください。準備した書き込みデータが格納されている領域の先頭アドレスを，汎用レジスタ ER0 に設定してください。このパラメータを FMPDR (フラッシュマルチパーパスデータデスティネーションエリアパラメータ) と呼びます。

(1) フラッシュマルチパーパスアドレスエリアパラメータ (FMPAR：CPU の汎用レジスタ ER1)
ユーザマット上の書き込み先の先頭アドレスを設定します。
フラッシュメモリ空間以外の領域のアドレスが設定されている場合，エラーとなります。
また，書き込み先の先頭アドレスは 128 バイト境界である必要があります。この境界条件になっていない場合も，エラーとなります。これらのエラーは FPFR パラメータの WA ビットに反映されます。

ビット	ビット名	R/W	説明
31～0	MOA31 ～ MOA0	R/W	ユーザマット上の書き込み先の先頭アドレスを格納します。ここで指定されたユーザマットの先頭アドレスから連続 128 バイトの書き込みが行われます。 よって, 指定する書き込み先の先頭アドレスは 128 バイト境界となり, MOA6～MOA0 は常に 0 になります。

(2) フラッシュマルチパーパスデータデスティネーションパラメータ (FMPDR：CPU の汎用レジスタ ER0)
ユーザマットに書き込むデータが格納されている領域の先頭アドレスを設定します。書き込みデータの格納先がフラッシュメモリ内の場合には，エラーとなります。このエラーは FPFR パラメータの WD ビットに反映されます。

ビット	ビット名	R/W	説明
31～0	MOD31 ～ MOD0	R/W	ユーザマットへの書き込みデータが格納されている領域の先頭アドレスを格納します。ここで指定された先頭アドレスから連続 128 バイトのデータが，ユーザマットに対して書き込まれます。

(3) フラッシュパス/フェイルパラメータ (FPFR：CPU の汎用レジスタ R0L)
書き込み処理結果の戻り値です。

ビット	ビット名	R/W	説明
7	–	–	リザーブビット： 値 0 が戻されます。
6	MD	R/W	書き込みモード関連設定エラー検出ビット エラープロテクト状態でないことのチェック結果を返します。 0：FLER 状態は正常 (FLER = 0) 1：FLER = 1 であり，書き込みできない状態
5	EE	R/W	書き込み実行時エラー検出ビット ユーザマットが消去されていないために，指定データを書き込めなかったり，ユーザブランチ処理から戻った時点で，フラッシュ関連レジスタの一部が書き換えられている場合に，本ビットには 1 が返されます。また，FMATS レジスタの値が H'AA となっており，ユーザブートマット選択状態のときに書き込みを実施しても，書き込み実行時エラーとなります。この場合は，ユーザマット/ユーザブートマットともに，書き換えられてはいません。ユーザブートマットの書き込みはブートモードまたはライタモードで実施してください。 0：書き込み処理は正常終了 1：書き込み処理が異常終了し，書き込み結果は保証できない
4	FK	R/W	フラッシュキーレジスタエラー検出ビット 書き込み処理開始前に FKEY レジスタの値をチェックした結果を戻します。 0：FKEY レジスタの設定は正常 (FKEY = H'5A) 1：FKEY レジスタの設定値エラー (FKEY は，H'5A 以外の値)
3	–	–	リザーブビット： 値 0 が戻されます
2	WD	R/W	ライトデータアドレス検出ビット 書き込みデータの格納先の先頭アドレスとして，以下の領域が指定された場合にはエラーとなります。 • 書き込み/消去プログラムがダウンロードされている内蔵 RAM 上のアドレス • フラッシュメモリ領域のアドレス 0：書き込みデータアドレス設定は正常値 1：書き込みデータアドレス設定が異常値

(次頁へ続く)

ビット	ビット名	R/W	説明
1	WA	R/W	ライトアドレスエラー検出ビット 書き込み先先頭アドレスとして，以下が指定された場合にはエラーとなります。 • フラッシュメモリの領域外の書き込み先アドレスの場合 • 指定されたアドレスが 128 バイト境界ではない (A6～A0 が 0 でない) 場合 0：書き込み先アドレス設定は正常値 1：書き込み先アドレス設定が異常値
0	SF	R/W	サクセス/フェイルビット 書き込み処理が正常に終了したかどうかを戻すビットです。 0：書き込みは正常終了 (エラー無し) 1：書き込みが異常終了 (エラーが発生している)

3.2.4 消去実行

フラッシュメモリの消去実行においては，ユーザマット上の消去ブロック番号をダウンロードした消去プログラムに渡すことが必要です。これを，FEBS パラメータ (汎用レジスタ ER0) に設定します。

0～15 のブロック番号から 1 ブロックを指定します。

(1) フラッシュイレースブロックセレクトパラメータ (FEBS：CPU の汎用レジスタ ER0)

消去ブロック番号を指定します。複数のブロック番号の指定はできません。

ビット	ビット名	R/W	説明
31～8	–	–	リザーブビット： 値 0 を設定してください。
7～0	EB7～EB0	R/W	イレースブロック 0～15 の範囲で消去ブロック番号を設定します。0 は EB0 ブロック，15 は EB15 ブロックに対応します。0～15 以外の設定ではエラーになります。

(2) フラッシュパス/フェイルパラメータ (FPFR：CPU の汎用レジスタ R0L)

消去処理結果の戻り値です。

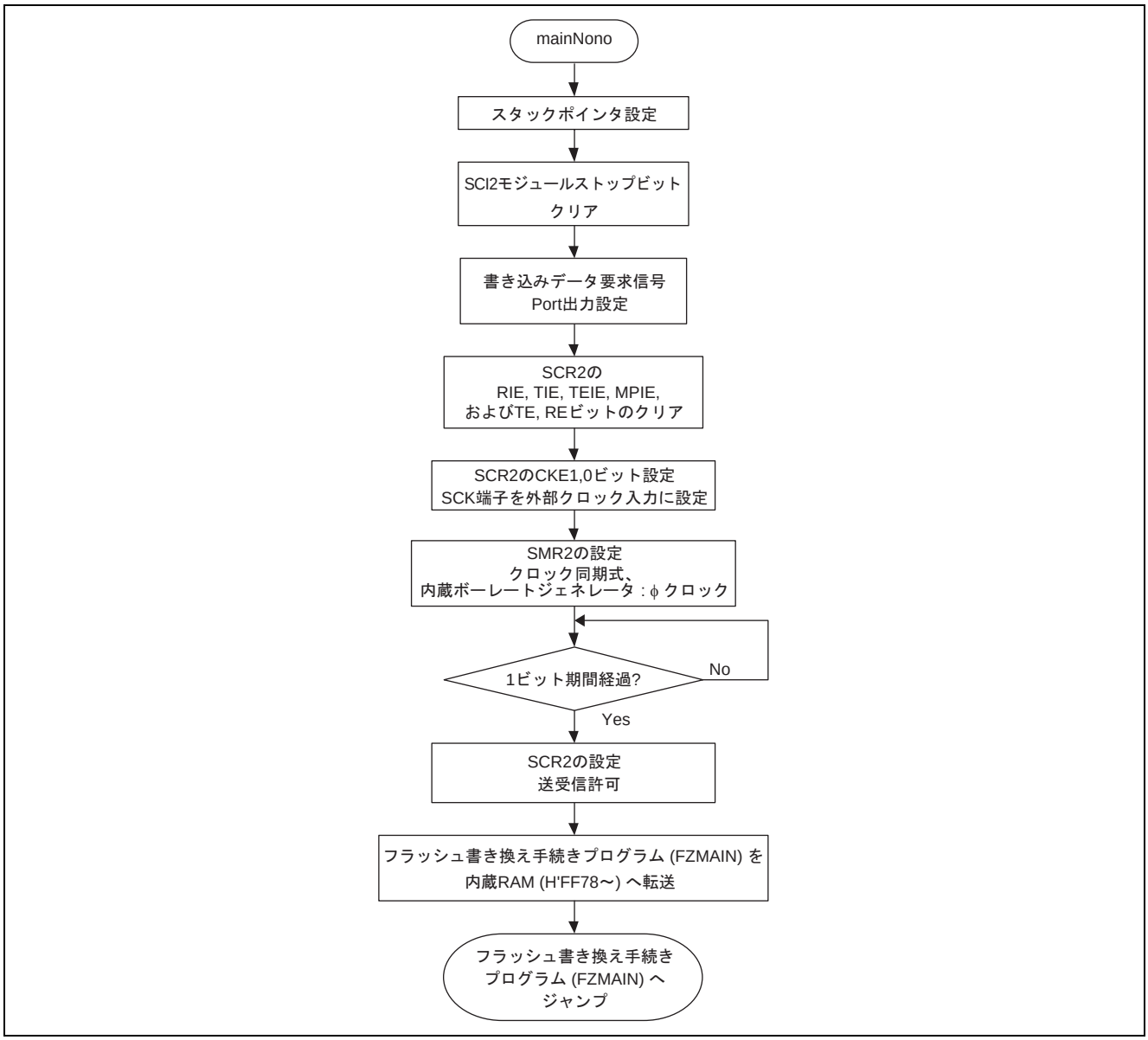
ビット	ビット名	R/W	説明
7	–	–	リザーブビット： 値 0 が戻されます。
6	MD	R/W	消去モード関連設定エラー検出ビット エラープロテクト状態でないことのチェック結果を返します。 0：FLER 状態は正常 (FLER = 0) 1：FLER = 1 であり，消去できない状態
5	EE	R/W	消去実行時エラー検出ビット ユーザマットが消去できなかったり，ユーザブランチ処理から戻った時点で，フラッシュ関連レジスタの一部が書き換えられている場合に，本ビットには 1 が返されます。また，FMATS レジスタの値が H'AA となっており，ユーザブートマット選択状態のときに消去を実施しても，消去実行時エラーとなります。この場合は，ユーザマット/ユーザブートマットともに，消去されていません。ユーザブートマットの消去はブートモードまたはライタモードで実施してください。 0：消去処理は正常終了 1：消去処理が異常終了し，消去結果は保証できない
4	FK	R/W	フラッシュキーレジスタエラー検出ビット 消去処理開始前に FKEY レジスタの値をチェックした結果を戻します。 0：FKEY レジスタの設定は正常 (FKEY = H'5A) 1：FKEY レジスタの設定値エラー (FKEY は，H'5A 以外の値)
3	EB	R/W	イレースブロックセレクトエラー検出ビット 指定された消去ブロック番号が，ユーザマットのブロック範囲内であるかのチェック結果です。 0：消去ブロック番号の設定は正常値 1：消去ブロック番号の設定が異常値
2,1	–	–	リザーブビット： 値 0 が戻されます。
0	SF	R/W	サクセス/フェイルビット 消去処理が正常に終了したかどうかを戻すビットです。 0：消去は正常終了 (エラー無し) 1：消去が異常終了 (エラーが発生している)

4. フローチャート

4.1 メイン処理 (ユーザブートプログラム)

起動後にスタックポインタ設定，SCI2 モジュールストップビット解除，ポートと SCI2 の初期設定，割り込み制御モードの設定を行い，フラッシュ書き込み/消去手続きプログラムを内蔵 RAM に転送後，フラッシュ書き込み/消去手続きプログラムにジャンプします。

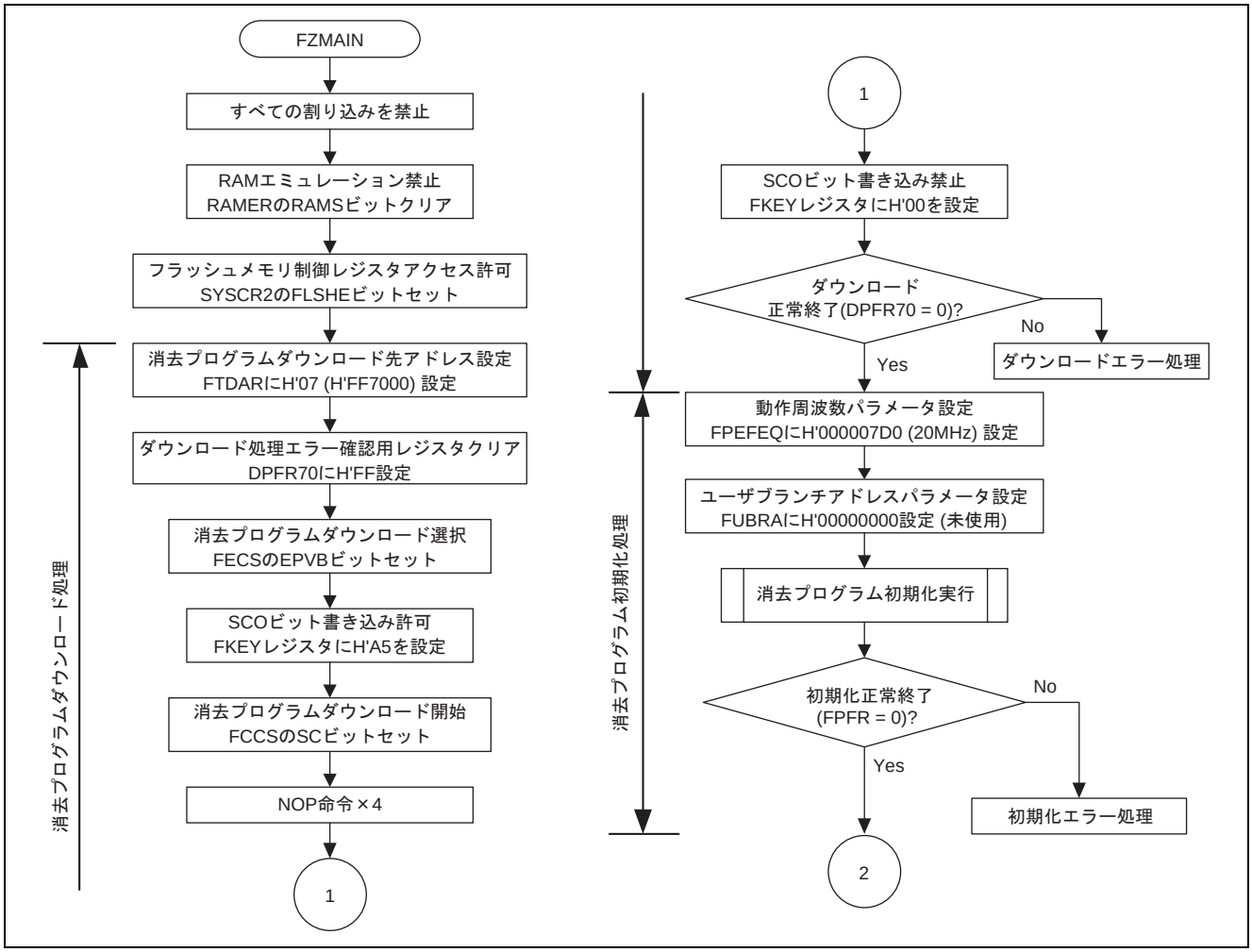
関数名	void main(void)
戻り値	–
引数	–
関数呼び出し	–
Section	MAIN

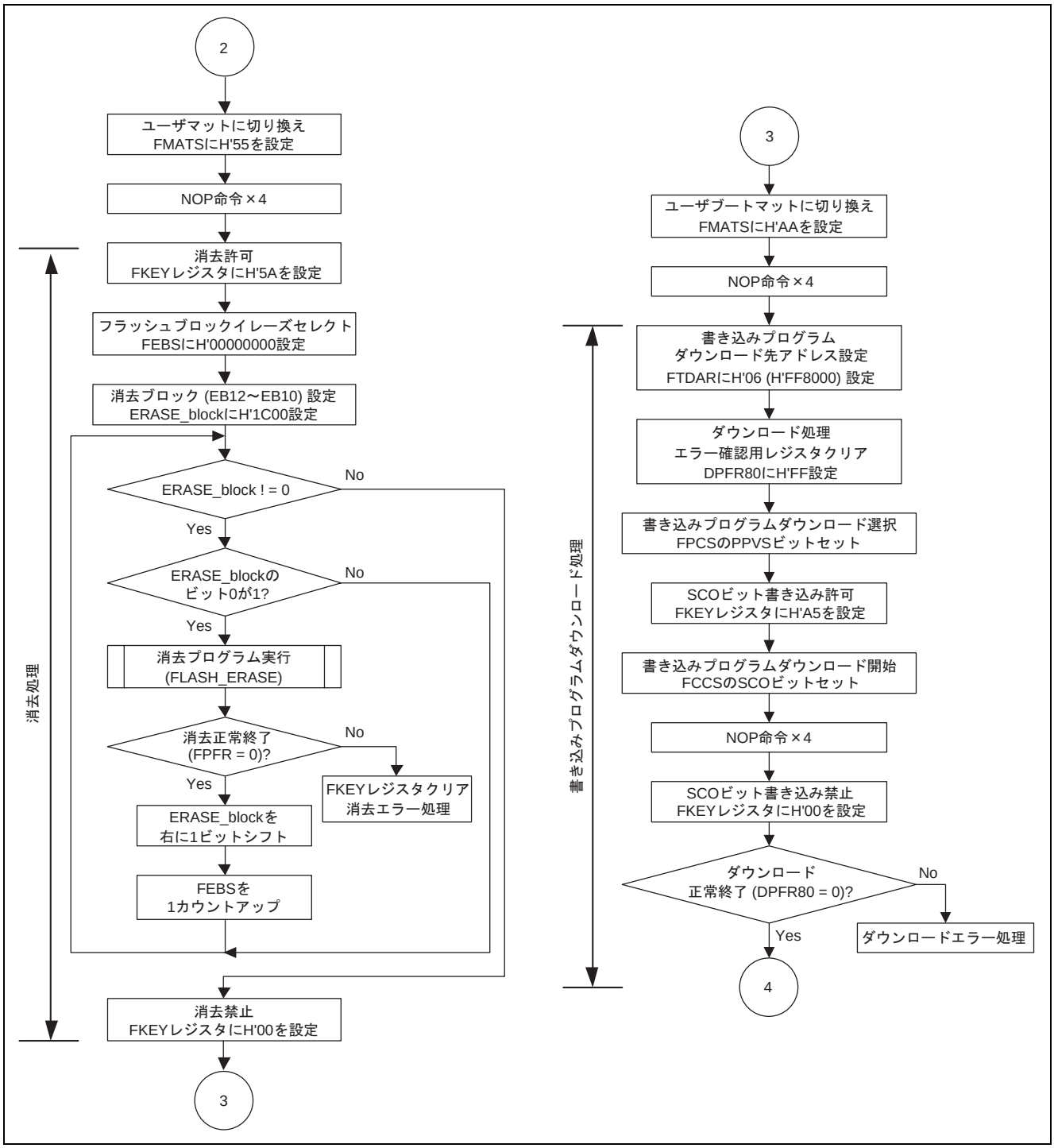


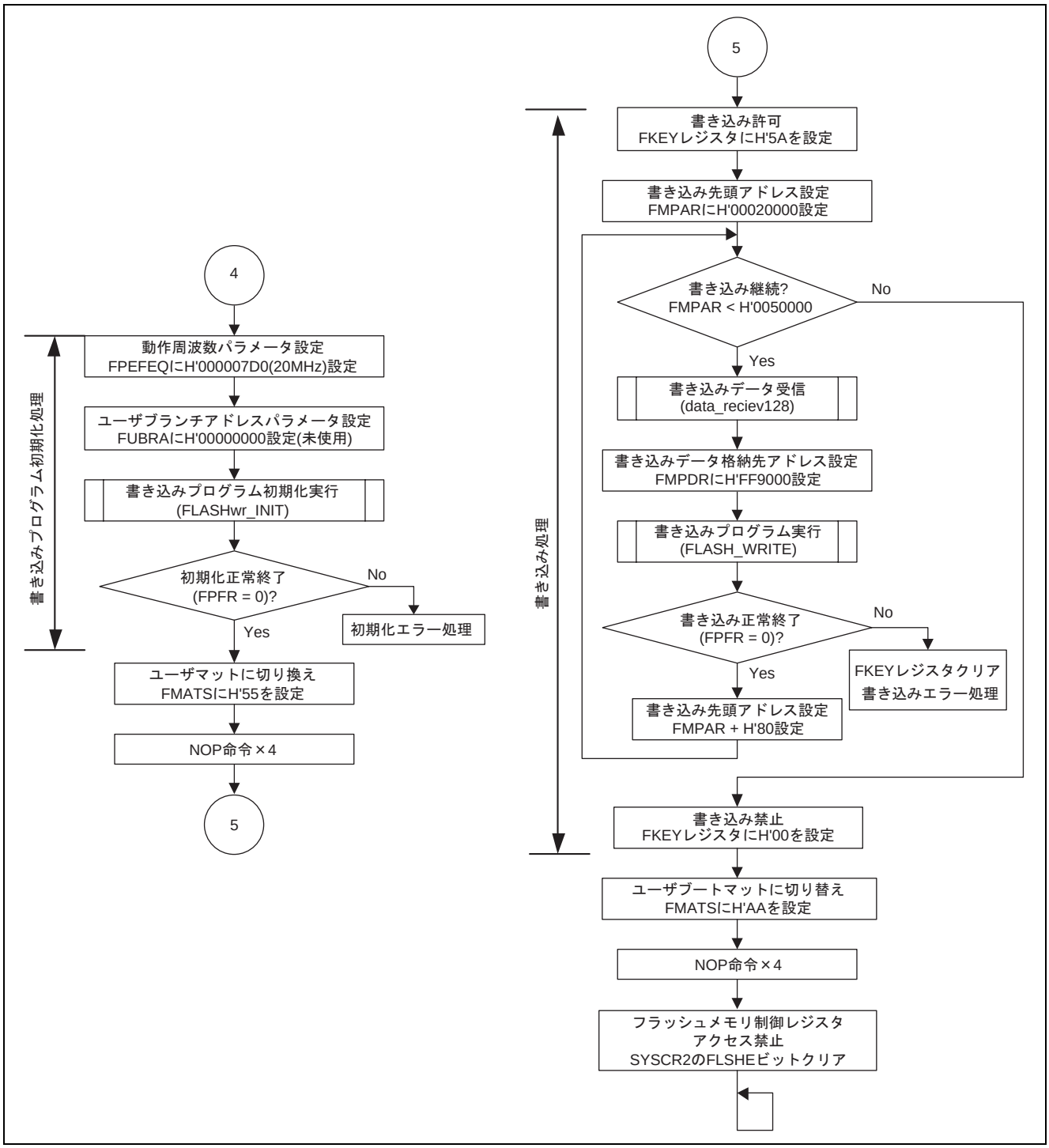
4.2 フラッシュ書き込み/消去手続きプログラム

書き込み/消去プログラムのダウンロード要求，書き込み/消去の手順，結果の判定および書き込みデータ要求，書き込みデータ受信などを内蔵 RAM 上で実行します。

関数名	void FZMAIN (void)
戻り値	–
引数	–
関数呼び出し	unsigned char FLASHer_INIT (unsigned long FPEFEQ, unsigned long FUBRA) unsigned char FLASH_ERASE (unsigned long FEBS) unsigned char FLASHwr_INIT (unsigned long FPEFEQ, unsigned long FUBRA) unsigned char FLASH_WRITE (unsigned long FMPDR, unsigned long FMPAR) void data_reciev128 (void)
Section	フラッシュメモリ格納エリア：FWRMAIN 内蔵 RAM 実行エリア：RWRMAIN(H'FF7800～)




2
ユーザマットに切り換え
FMATSにH'55を設定
NOP命令×4
消去許可
FKEYレジスタにH'5Aを設定
フラッシュブロックイレーズセレクト
FEBSにH'00000000設定
消去ブロック (EB12～EB10) 設定
ERASE_blockにH'1C00設定
ERASE_block ! = 0
No
Yes
ERASE_blockの
ビット0が1?
No
Yes
消去プログラム実行
(FLASH_ERASE)
消去正常終了
(FPFR = 0)?
No
Yes
FKEYレジスタクリア
消去エラー処理
ERASE_blockを
右に1ビットシフト
FEBSを
1カウントアップ
消去禁止
FKEYレジスタにH'00を設定
3
消去処理
3
ユーザブートマットに切り換え
FMATSにH'AAを設定
NOP命令×4
書き込みプログラム
ダウンロード先アドレス設定
FTDARにH'06 (H'FF8000) 設定
ダウンロード処理
エラー確認用レジスタクリア
DPFR80にH'FF設定
書き込みプログラムダウンロード選択
FPCSのPPVSビットセット
SCOビット書き込み許可
FKEYレジスタにH'A5を設定
書き込みプログラムダウンロード開始
FCCSのSCOビットセット
NOP命令×4
SCOビット書き込み禁止
FKEYレジスタにH'00を設定
ダウンロード
正常終了 (DPFR80 = 0)?
No
Yes
ダウンロードエラー処理
4
書き込みプログラムダウンロード処理


4
動作周波数パラメータ設定
FPEFEQにH'000007D0(20MHz)設定
ユーザブランチアドレスパラメータ設定
FUBRAにH'00000000設定(未使用)
書き込みプログラム初期化実行
(FLASHwr_INIT)
初期化正常終了
(FPFR = 0)?
No
初期化エラー処理
Yes
ユーザマットに切り換え
FMATSにH'55を設定
NOP命令×4
5
書き込みプログラム初期化処理
5
書き込み許可
FKEYレジスタにH'5Aを設定
書き込み先頭アドレス設定
FMPARにH'00020000設定
書き込み継続?
FMPAR < H'0050000
No
Yes
書き込みデータ受信
(data_reciev128)
書き込みデータ格納先アドレス設定
FMPDRにH'FF9000設定
書き込みプログラム実行
(FLASH_WRITE)
書き込み正常終了
(FPFR = 0)?
No
FKEYレジスタクリア
書き込みエラー処理
Yes
書き込み先頭アドレス設定
FMPAR + H'80設定
書き込み処理
書き込み禁止
FKEYレジスタにH'00を設定
ユーザブートマットに切り替え
FMATSにH'AAを設定
NOP命令×4
フラッシュメモリ制御レジスタ
アクセス禁止
SYSCR2のFLSHEビットクリア

4.3 フラッシュメモリ書き込み初期化処理 (ダミー)

フラッシュメモリ書き込み初期化処理は, 内蔵されたプログラムをフラッシュ書き込み/消去手続きプログラムによって内蔵 RAM にダウンロードされ，内蔵 RAM 上で実行します。

本アプリケーション例では手続きプログラムで指定アドレスにダウンロードした後，フラッシュメモリ書き込み初期化処理関数を関数名 (FLASHwr_INIT) で呼び出し可能にするため, あらかじめダミーで関数名のみ宣言し実行アドレスを指定しています。

ダミー宣言のためフラッシュメモリ (ユーザブートマット) 上には 2 バイトの RTS 命令のみ格納されます。なお，フラッシュメモリから内蔵 RAM への転送は行いません。

関数名	unsigned char FLASHwr_INIT(unsigned long FPEFEQ, unsigned long FUBRA)
戻り値	• フラッシュパスフェイルパラメータ：FPFR(R0L) 書き込み初期化結果の戻り値です。正常終了時に戻り値は H'00 です。
引数	• フラッシュプログラムイレース周波数コントロール：FPEFEQ(ER0) CPU の動作周波数を設定します。MHz 単位で表現した動作周波数を小数点第 3 位で四捨五入し，小数点第 2 位までとする。この値を 100 倍した値を 16 進数に変換し設定する。 本アプリケーション例では，20.0MHz 動作ですので，20000 = H'07D0 を設定します。 • フラッシュユーザブランチアドレスセットパラメータ：FUBRA(ER1) ユーザブランチ先のアドレスを設定するパラメータです。 本アプリケーション例では，必要ないので，0 番地 (H'00000000) を設定します。
関数呼び出し	–
Section	フラッシュ格納エリア：FWRINI 内蔵 RAM 実行エリア：RWRINI (H'FF7020～)

4.4 フラッシュメモリ書き込み処理 (ダミー)

フラッシュメモリ書き込み処理は, 内蔵されたプログラムをフラッシュ書き込み/消去手続きプログラムによって内蔵 RAM にダウンロードされ，内蔵 RAM 上で実行します。

本アプリケーション例では手続きプログラムで指定アドレスにダウンロードした後，フラッシュメモリ書き込み処理関数を関数名 (FLASH_WRITE) で呼び出し可能にするため, あらかじめダミーで関数名のみ宣言し実行アドレスを指定しています。

ダミー宣言のためフラッシュメモリ (ユーザブートマット) 上には 2 バイトの RTS 命令のみ格納されます。なお，フラッシュメモリから内蔵 RAM への転送は行いません。

関数名	unsigned char FLASH_WRITE(unsigned long FMPDR, unsigned long FMPAR)
戻り値	• フラッシュパスフェイルパラメータ：FPFR(R0L) 書き込み処理結果の戻り値です。正常終了時に戻り値は H'00 です。
引数	• フラッシュマルチパーパスデータデスティネーションパラメータ : FMPDR (ER0) フラッシュメモリに書き込むデータが格納されている領域の先頭アドレスを設定します。 本アプリケーション例では，内蔵 RAM 領域 H'FF9000 から 128 バイトですので，H'FF9000 を設定します。 • フラッシュマルチパーパスアドレスエリアパラメータ：FMPAR (ER1) フラッシュメモリ書き込み先の先頭アドレスを設定します。設定アドレスは 128 バイト境界である必要があります。
関数呼び出し	–
Section	フラッシュ格納エリア：FWRIT 内蔵 RAM 実行エリア：RWRIT (H'FF7010～)

4.5 フラッシュメモリ消去初期化処理 (ダミー)

フラッシュメモリ消去初期化処理は, 内蔵されたプログラムをフラッシュ書き込み/消去手続きプログラムによって内蔵 RAM にダウンロードされ，内蔵 RAM 上で実行します。

本アプリケーション例では手続きプログラムで指定アドレスにダウンロードした後，フラッシュメモリ消去初期化処理関数を関数名 (FLASHer_INIT) で呼び出し可能にするため，あらかじめダミーで関数名のみ宣言し実行アドレスを指定しています。

ダミー宣言のためフラッシュメモリ (ユーザブートマット) 上には 2 バイトの RTS 命令のみ格納されます。なお，フラッシュメモリから内蔵 RAM への転送は行いません。

関数名	unsigned char FLASHer_INIT(unsigned long FPEFEQ, unsigned long FUBRA)
戻り値	• フラッシュパスフェイルパラメータ：FPFR(R0L) 消去初期化処理結果の戻り値です。正常終了時に戻り値は H'00 です。
引数	• フラッシュプログラムイレース周波数コントロール：FPEFEQ(ER0) CPU の動作周波数を設定します。MHz 単位で表現した動作周波数を小数点第 3 位で四捨五入し，小数点第 2 位までとする。この値を 100 倍した値を 16 進数に変換し設定する。 本アプリケーション例では，20.0MHz 動作ですので，20000 = H'07D0 を設定します。 • フラッシュユーザブランチアドレスセットパラメータ：FUBRA(ER1) ユーザブランチ先のアドレスを設定するパラメータです。 本アプリケーション例では，必要ないので，0 番地 (H'00000000) を設定します。
関数呼び出し	–
Section	フラッシュ格納エリア：FERINI 内蔵 RAM 実行エリア：RERINI (H'FF7020～)

4.6 フラッシュメモリ消去処理 (ダミー)

フラッシュメモリ消去処理は, 内蔵されたプログラムをフラッシュ書き込み/消去手続きプログラムによって内蔵 RAM にダウンロードされ，内蔵 RAM 上で実行します。

本アプリケーション例では手続きプログラムで指定アドレスにダウンロードした後，フラッシュメモリ消去処理関数を関数名 (FLASH_ERASE) で呼び出し可能にするため, あらかじめダミーで関数名のみ宣言し実行アドレスを指定しています。

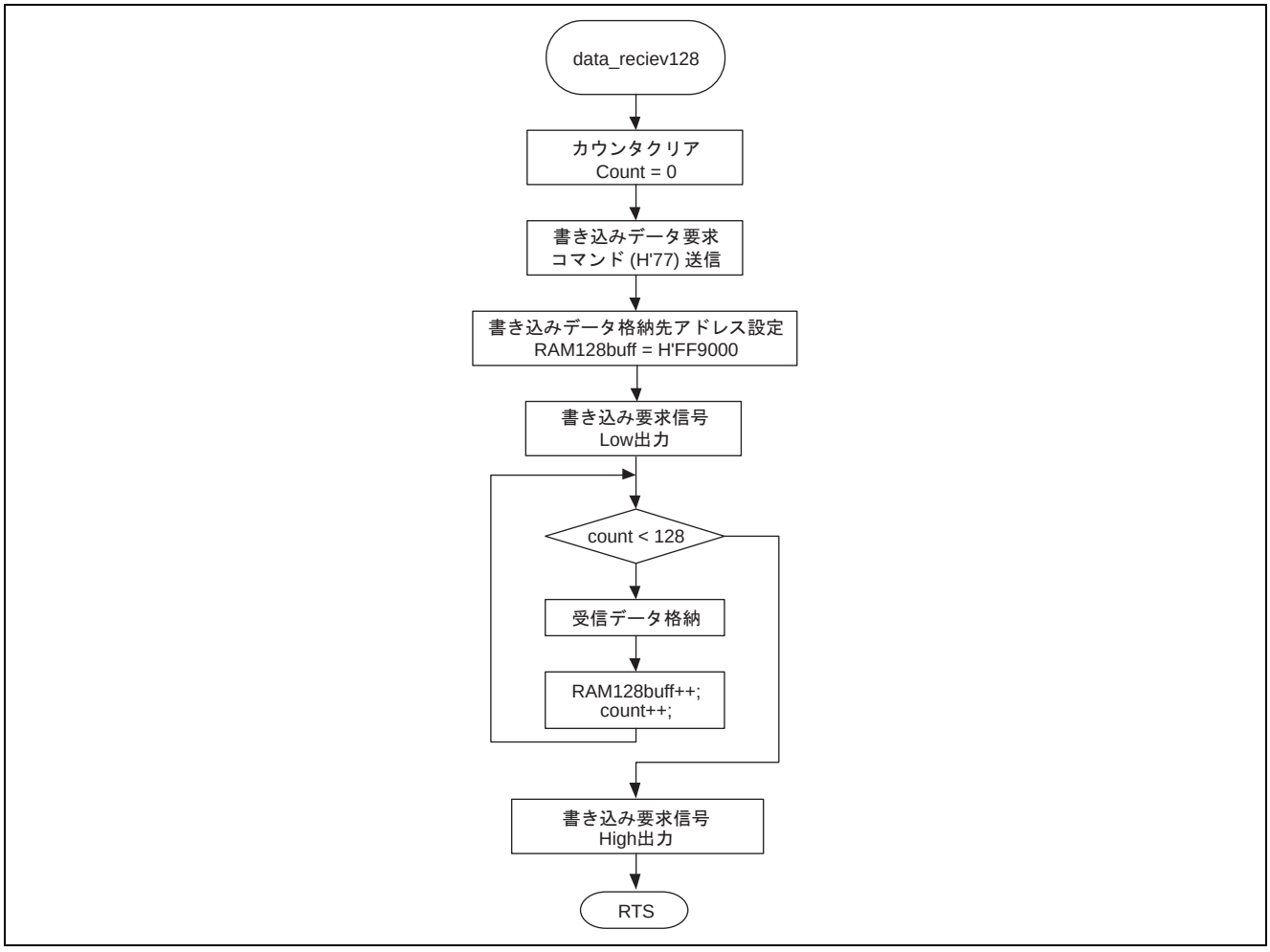
ダミー宣言のためフラッシュメモリ (ユーザブートマット) 上には 2 バイトの RTS 命令のみ格納されます。なお，フラッシュメモリから内蔵 RAM への転送は行いません。

関数名	unsigned char FLASH_ERASE(unsigned long FEBS)
戻り値	• フラッシュパスフェイルパラメータ：FPFR(R0L) 消去処理結果の戻り値です。正常終了時に戻り値は H'00 です。
引数	• フラッシュイレースブロックセレクトパラメータ：FEBS (ER0) 消去ブロック番号を設定します。複数のブロック番号は指定できません。 本アプリケーション例では，EB12～EB10 を消去します。 EB12 は H'0000000C,EB11 は H'0000000B,EB10 は H'0000000A を設定します。
関数呼び出し	–
Section	フラッシュ格納エリア：FERAS 内蔵 RAM 実行エリア：RERAS (H'FF7010～)

4.7 書き込みデータ受信処理

フラッシュメモリ書き込み処理前に書き込みデータ要求信号を書き込みツールに送信し，128 バイトの書き込みデータを SCI2 (クロック同期) で受信しながら内蔵 RAM に格納します。

関数名	void data_reciev128(void)
戻り値	–
引数	–
関数呼び出し	–
Section	フラッシュ書き込み/消去手続きプログラムと同じセクション (FWRMAIN) で FZMAIN の後ろに配置します。実行エリアも同じです。



5. メモリマップ

本アプリケーションプログラムのメモリマップを以下に示します。

フラッシュメモリ (ユーザブートマット)		Section	内容
H'000000～		VECT	ベクタテーブル
H'000400～		MAIN	4.1 章参照
H'000800～		FWRMAIN	4.2 章，4.7 章参照
		FWRINI	4.3 章参照
		FWRIT	4.4 章参照
		FERINI	4.5 章参照
		FERAS	4.6 章参照
フラッシュメモリ (ユーザマット)		Section	内容
H'020000～H'04FFFF		DATA	データエリア (EB10～EB12)
内蔵 RAM		Section	内容
H'FF7000～H'FF77FF		–	消去/書き込みプログラムダウンロード先
	H'FF7010	RERAS/RWRIT	4.6 章参照，4.4 章参照
	H'FF7020	RERINI/RWRINI	4.5 章参照，4.3 章参照
H'FF7800～		RWRMAIN	4.2 章参照

6. プログラムソース

6.1 メイン処理 (ユーザアプリケーション) ソース

内蔵 RAM へのジャンプソースとセクションラベルを合わせて示します。

```
/*****************************/
/*  main                     */
/*****************************/
#pragma entry main(sp=0xFFEFB0)          /* スタックポインタ設定                  */
void main(void){
   unsigned long  i=0;
   *MSTPCRA = 0x3f;
   *MSTPCRB = 0x5f;                      /* SCI2 モジュールストップビットクリア   */
   *MSTPCRC = 0xff;
   *PJDDR = 0xFF;                        /* 書き込み要求信号用ポート初期設定      */
   *PJDR = 0x01;                         /* 書き込み要求信号 High 出力            */
   *SCR2  = 0x00;                    /* RIE,TIE,TEIE,MPIE,TE,RE ビットのクリア   */
   *SCR2  = 0x03;                        /* 外部クロック選択                      */
   *SMR2  = 0x80;                        /* クロック同期式                        */
   *SCMR2 = 0xF2;                    /* LSB ファーストで送受信，通常のクロック同期式 */
   for( i=0 ; i<20 ; i++ );              /* wait                                  */
   *SCR2  = 0x13;                        /* RE セット                             */
   for(src=_FWRMAIN_BGN, dst=_RWRMAIN_RAM; src<=_FWRMAIN_END; src++, dst++) {
      *dst = *src;                       /* 内蔵 RAM へ転送                       */
   }
   jmp_RAM();                            /* 受信データ格納                        */
   while(1);
}
/*****************************/
/* 内蔵 RAM へのジャンプ     */
/*****************************/
#pragma inline_asm(jmp_RAM)
void  jmp_RAM(void){
   MOV.L #H'FFA000,ER0
   JMP   @ER0
   NOP
}
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;  セクションラベルテーブル
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
         .SECTION     FWRMAIN,          CODE,ALIGN=2
         .SECTION     RWRMAIN,          CODE,ALIGN=2
         .SECTION     C,DATA,ALIGN=2
         .EXPORT __FWRMAIN_BGN
         .EXPORT __FWRMAIN_END
         .EXPORT __RWRMAIN_RAM
__FWRMAIN_BGN  .data.l (STARTOF FWRMAIN)
__FWRMAIN_END  .data.l (STARTOF FWRMAIN) + (SIZEOF FWRMAIN)
__RWRMAIN_RAM  .data.l (STARTOF RWRMAIN)
   .END
```

6.2 フラッシュ書き込み／消去手続きプログラムソース

フラッシュ書き込み／消去手続き処理で使用する，書き込みデータ受信処理を合わせて示します。

```
#define DPFR70 (*(unsigned char  *)0xff7000)
                                    /* ダウンロード処理エラー確認用レジスタ(DPFR)   */
#pragma inline_asm(NOP4)
static void NOP4(void){                 /* NOP 命令４回                          */
   NOP
   NOP
   NOP
   NOP
}
void FZMAIN(void){                      /* 0xFF2000～                            */
   unsigned long FPEFEQ;                /* ER0                                   */
   unsigned long FUBRA;                 /* ER1                                   */
   unsigned char FPFR;                  /* R0L                                   */
   unsigned long FEBS;                  /* ER0                                   */
   unsigned long FMPAR;                 /* ER1                                   */
   unsigned long FMPDR;                 /* ER0                                   */
   *RAMER=0;                            /* エミュレーション非選択                */
   *SYSCR2=0x08;                    /* フラッシュメモリ制御レジスタアクセス許可      */
/*  消去プログラムダウンロード処理      */
   *FTDAR = 0x07;                       /* ダウンロード先アドレス設定 (H'FF7000) */
   DPFR70 = 0xFF;           /* ダウンロード処理エラー確認用レジスタ (DPFR)クリア    */
   *FECS = 0x01;                        /* 消去プログラム選択 (EPVB set)         */
   *FKEY = 0xA5;                        /* SCO ビット書き込み許可                */
   *FCCS |= 0x01;                   /* 消去プログラムダウンロード要求 (SCO set)     */
   NOP4();                              /* NOP×4                                 */
   *FKEY = 0x00;                        /* FKEY clear                            */
   if(DPFR70 != 0x00){                  /* ダウンロードエラー?                   */
      goto end_p;
   }
/*  消去初期化処理        */
   FPEFEQ = 0x000007D0;     /* 動作周波数パラメータ設定 20MHz H'07D0 => D'2000   */
   FUBRA = 0x00000000;                  /* ユーザブランチアドレスパラメータ設定  */
   FPFR = FLASHer_INIT(FPEFEQ,FUBRA);
                                    /* FPFR-->R0L,FPEFEQ-->ER0,FUBRA-->ER1     */
   if(FPFR != 0x00){                    /* 初期設定エラー?                       */
      goto end_p;
   }
/*  マット切り換え   */
   *FMATS = 0x55;                       /* ユーザマット選択                      */
   NOP4();                              /* NOP×4                                 */
/*  消去処理     */
   *FKEY = 0x5A;                        /* FKEY set                              */
   FEBS=0x00000000;
   ERASE_block = 0x1C00;                /* EB12～EB10 erase                      */
   while(ERASE_block != 0x0000){        /* ブロック消去ルーチン                  */
      if((ERASE_block & 0x0001)==0x0001){
         FPFR = FLASH_ERASE(FEBS);      /* FPFR-->R0L, FEBS-->ER0                */
         if(FPFR != 0x00){              /* 消去エラー?                           */
            goto end_p;
         }
      }
      ERASE_block = (ERASE_block>>1);   /* 消去ブロック指定レジスタシフト        */
```

```
      FEBS++;                              /* フラッシュイレーズセレクトパラメータ    */
   }
   *FKEY = 0x00;                           /* FKEY clear                          */
/*  マット切り換え   */
   *FMATS = 0xAA;                          /* ユーザブートマット選択              */
   NOP4();                                 /* NOP×4                               */
/*  書き込みプログラムダウンロード処理   */
   *FTDAR = 0x07;                          /* ダウンロード先アドレス設定 (H'FF7000) */
   DPFR70 = 0xFF;            /* ダウンロード処理エラー確認用レジスタ (DPFR) クリア   */
   *FPCS = 0x01;                           /* 書き込みプログラム選択 (PPVS set)   */
   *FKEY = 0xA5;                           /* 書き込みプログラムダウンロード許可  */
   *FCCS |= 0x01;                          /* ダウンロード要求 (SCO set)          */
   NOP4();                                 /* NOP×4                               */
   *FKEY = 0x00;                           /* FKEY clear                          */
   if(DPFR70 != 0x00){                     /* ダウンロードエラー?                 */
      goto end_p;
   }
/*  書き込み初期化処理          */
   FPEFEQ = 0x000007D0;      /* 動作周波数パラメータ設定 20MHz H'07D0 => D'2000   */
   FUBRA = 0x00000000;                     /* ユーザブランチアドレスパラメータ設定 */
   FPFR = FLASHwr_INIT(FPEFEQ,FUBRA);
                                  /* FPEFEQ-->ER0,FUBRA-->R5,ERR_CHECK-->R0 */
   if(FPFR != 0x00){                       /* 初期設定エラー?                     */
      goto end_p;
   }
/*  マット切り換え   */
   *FMATS = 0x55;                          /* ユーザマット選択                    */
   NOP4();                                 /* NOP×4                               */
/*  書き込み処理         */
   *FKEY = 0x5A;                           /* FKEY set                            */
   FMPAR = 0x00020000;                     /* 書き込み先頭アドレス設定            */
   while(FMPAR < 0x00050000){              /* 書き込み終了?                       */
      data_reciev128();                    /* 書き込みデータ受信ルーチン          */
      FMPDR=0xFFFF9000;
      FPFR = FLASH_WRITE(FMPDR,FMPAR);
                                  /* FMPDR-->R4,FMPAR-->R5,ERR_CHECK-->R0   */
      if(FPFR != 0x00){                    /* 書き込みエラー?                     */
         goto end_p;
      }
      FMPAR+=0x80;
   }
   *FKEY = 0x00;                           /* FKEY clear                          */
end_p:                                     /* エラー発生時の飛び先                */
/*  マット切り換え   */
   *FMATS = 0xAA;                          /* ユーザブートマット選択              */
   NOP4();                                 /* NOP×4                               */
   *SYSCR2=0x00;
   while(1);
}

void data_reciev128(void){
   unsigned char count=0;                  /* 受信データカウンタ                  */
   unsigned char *RAM128buff;              /* 書き込みデータ格納先内蔵 RAM アドレス */
RAM128buff=(unsigned char *)0xFFFF9000;
*PJDR &= 0xFE;                             /* 書き込みデータ要求信号 Low 出力     */
```

```
    while(count < 128){
    while((*SSR2 & 0x40)!=0x40);
       *SSR2 &= 0xbf;
       *RAM128buff = *RDR2;              /* 受信データ格納                    */
       RAM128buff++;
       count++;
    }
    *PJDR |= 0x01;                       /* 書き込みデータ要求信号 High 出力  */
}
```

6.3 フラッシュメモリ消去初期化処理ダミーソース

4.5 章参照。

```
unsigned char FLASHer_INIT(unsigned long FPEFEQ, unsigned long FUBRA){}
```

6.4 フラッシュメモリ消去処理ダミーソース

4.6 章参照。

```
unsigned char FLASH_ERASE(unsigned long FEBS){}
```

6.5 フラッシュメモリ書き込み初期化処理ダミーソース

4.3 章参照。

```
unsigned char FLASHwr_INIT(unsigned long FPEFEQ,unsigned long FUBRA){}
```

6.6 フラッシュメモリ書き込み処理ダミーソース

4.4 章参照。

```
unsigned char FLASH_WRITE(unsigned long FMPDR, unsigned long FMPAR){}
```

7. 付録

I/O 定義ヘッダファイル (ioaddrs.h)

```
/************************/
/*    System            */
/************************/
#define SYSCR2 (volatile char*)(0xfffde2)
#define SBYCR (volatile char*)(0xfffde4)
#define SYSCR (volatile char*)(0xfffde5)
#define SCKCR (volatile char*)(0xfffde6)
#define MDCR (volatile char*)(0xfffde7)
#define MSTPCRA (volatile char*)(0xfffde8)
#define MSTPCRB (volatile char*)(0xfffde9)
#define MSTPCRC (volatile char*)(0xfffdea)
#define LPWRCR (volatile char*)(0xfffdec)

/************************/
/*      INTC            */
/************************/
#define ISCRH (volatile char*)(0xfffe12)
#define ISCRL (volatile char*)(0xfffe13)
#define IER (volatile char*)(0xfffe14)
#define ISR (volatile char*)(0xfffe15)
#define IPRA (volatile char*)(0xfffec0)
#define IPRB (volatile char*)(0xfffec1)
#define IPRC (volatile char*)(0xfffec2)
#define IPRD (volatile char*)(0xfffec3)
#define IPRE (volatile char*)(0xfffec4)
#define IPRF (volatile char*)(0xfffec5)
#define IPRG (volatile char*)(0xfffec6)
#define IPRH (volatile char*)(0xfffec7)
#define IPRI (volatile char*)(0xfffec8)
#define IPRJ (volatile char*)(0xfffec9)
#define IPRK (volatile char*)(0xfffeca)
#define IPRL (volatile char*)(0xfffecb)
#define IPRM (volatile char*)(0xfffecc)
#define IPRO (volatile char*)(0xfffece)

/************************/
/* SCI2                 */
/************************/
#define SMR2 (volatile char*)(0xffff88)
#define BRR2 (volatile char*)(0xffff89)
#define SCR2 (volatile char*)(0xffff8a)
#define TDR2 (volatile char*)(0xffff8b)
#define SSR2 (volatile char*)(0xffff8c)
#define RDR2 (volatile char*)(0xffff8d)
#define SCMR2 (volatile char*)(0xffff8e)

/************************/
/*    I/O Port          */
/************************/
#define P1DDR (volatile char*)(0xfffe30)
#define P2DDR (volatile char*)(0xfffe31)
#define P3DDR (volatile char*)(0xfffe32)
#define P5DDR (volatile char*)(0xfffe34)
#define P7DDR (volatile char*)(0xfffe36)
```

```
#define P3ODR (volatile char*)(0xfffe46)

#define P1DR (volatile char*)(0xffff00)
#define P2DR (volatile char*)(0xffff01)
#define P3DR (volatile char*)(0xffff02)
#define P5DR (volatile char*)(0xffff04)
#define P7DR (volatile char*)(0xffff06)

#define PORT1 (volatile char*)(0xffffb0)
#define PORT2 (volatile char*)(0xffffb1)
#define PORT3 (volatile char*)(0xffffb2)
#define PORT4 (volatile char*)(0xffffb3)
#define PORT5 (volatile char*)(0xffffb4)
#define PORT7 (volatile char*)(0xffffb6)
#define PORT9 (volatile char*)(0xffffb8)

#define PADDR (volatile char*)(0xfffe39)
#define PADR (volatile char*)(0xffff09)
#define PORTA (volatile char*)(0xffffb9)
#define PAPCR (volatile char*)(0xfffe40)
#define PAODR (volatile char*)(0xfffe47)

#define PBDDR (volatile char*)(0xfffe3a)
#define PBDR (volatile char*)(0xffff0a)
#define PORTB (volatile char*)(0xffffba)
#define PBPCR (volatile char*)(0xfffe41)

#define PCDDR (volatile char*)(0xfffe3b)
#define PCDR (volatile char*)(0xffff0b)
#define PORTC (volatile char*)(0xffffbb)
#define PCPCR (volatile char*)(0xfffe42)

#define PDDDR (volatile char*)(0xfffe3c)
#define PDDR (volatile char*)(0xffff0c)
#define PORTD (volatile char*)(0xffffbc)
#define PDPCR (volatile char*)(0xfffe43)

#define PEDDR (volatile char*)(0xfffe3d)
#define PEDR (volatile char*)(0xffff0d)
#define PORTE (volatile char*)(0xffffbd)
#define PEPCR (volatile char*)(0xfffe44)

#define PFDDR (volatile char*)(0xfffe3e)
#define PFDR (volatile char*)(0xffff0e)
#define PORTF (volatile char*)(0xffffbe)

#define PGDDR (volatile char*)(0xfffe3f)
#define PGDR (volatile char*)(0xffff0f)
#define PORTG (volatile char*)(0xffffbf)

#define PHDDR (volatile char*)(0xfffa10)
#define PJDDR (volatile char*)(0xfffa11)
#define PHDR (volatile char*)(0xfffa12)
#define PJDR (volatile char*)(0xfffa13)
#define PORTH (volatile char*)(0xfffa14)
#define PORTJ (volatile char*)(0xfffa15)
```

```
/************************/
/*    ROM               */
/************************/
#define RAMER (volatile char*)(0xfffedb)
#define FCCS (volatile char*)(0xffffa4)
#define FPCS (volatile char*)(0xffffa5)
#define FECS (volatile char*)(0xffffa6)
#define FKEY (volatile char*)(0xffffa8)
#define FMATS (volatile char*)(0xffffa9)
#define FTDAR (volatile char*)(0xffffaa)
#define FVACR (volatile char*)(0xffffab)
#define FVADRR (volatile char*)(0xffffac)
#define FVADRE (volatile char*)(0xffffad)
#define FVADRH (volatile char*)(0xffffae)
#define FVADRL (volatile char*)(0xffffaf)
```

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2005.02.18	–	初版発行

安全設計に関するお願い

1. 弊社は品質、信頼性の向上に努めておりますが、半導体製品は故障が発生したり、誤動作する場合があります。弊社の半導体製品の故障又は誤動作によって結果として、人身事故、火災事故、社会的損害などを生じさせないような安全性を考慮した冗長設計、延焼対策設計、誤動作防止設計などの安全設計に十分ご留意ください。

本資料ご利用に際しての留意事項

1. 本資料は、お客様が用途に応じた適切なルネサス テクノロジ製品をご購入いただくための参考資料であり、本資料中に記載の技術情報についてルネサス テクノロジが所有する知的財産権その他の権利の実施、使用を許諾するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例の使用に起因する損害、第三者所有の権利に対する侵害に関し、ルネサス テクノロジは責任を負いません。
3. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他全ての情報は本資料発行時点のものであり、ルネサス テクノロジは、予告なしに、本資料に記載した製品または仕様を変更することがあります。ルネサス テクノロジ半導体製品のご購入に当たりましては、事前にルネサス テクノロジ、ルネサス販売または特約店へ最新の情報をご確認頂きますとともに、ルネサス テクノロジホームページ(http://www.renesas.com)などを通じて公開される情報に常にご注意ください。
4. 本資料に記載した情報は、正確を期すため、慎重に制作したものですが万一本資料の記述誤りに起因する損害がお客様に生じた場合には、ルネサス テクノロジはその責任を負いません。
5. 本資料に記載の製品データ、図、表に示す技術的な内容、プログラム及びアルゴリズムを流用する場合は、技術内容、プログラム、アルゴリズム単位で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。ルネサス テクノロジは、適用可否に対する責任は負いません。
6. 本資料に記載された製品は、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。本資料に記載の製品を運輸、移動体用、医療用、航空宇宙用、原子力制御用、海底中継用機器あるいはシステムなど、特殊用途へのご利用をご検討の際には、ルネサス テクノロジ、ルネサス販売または特約店へご照会ください。
7. 本資料の転載、複製については、文書によるルネサス テクノロジの事前の承諾が必要です。
8. 本資料に関し詳細についてのお問い合わせ、その他お気付きの点がございましたらルネサス テクノロジ、ルネサス販売または特約店までご照会ください。